July 1988
$4.50 U.S.
$5.95 Canada

ST Applications

THE ATARI ST JOURNAL

LateST News * Software Reviews * BeST and WorST
The Making Of Stocks & Bonds * Super Slot Machine
APL * Picshell Menu * Add-On For 3D Maze Run

07
0 74470 19763 8

Now Includes: ST™ BUSINESS

ST Applications
THE ATARI ST JOURNAL

Editor and Publisher: William Petry

Assistant Editors: Sharon Richmond
Nancy Koepp

Photographer: Martin Majano

Subscriptions: Amy McLintock

Assistant to the Publisher: Michelle Stevens

ST Applications- *The Atari ST Journal* (ISSN 0887-3755), is copyright © 1988, and published monthly by **ST Applications,** 9890 River Road, P.O. Box 980, Forestville, CA 95436 USA. Phone (707) 887-7879. Editorial office at 9890 River Road, Forestville, CA 95436. Hours: 8:30 A.M.- 5:00 P.M., Pacific Time, Monday thru Friday.

Subscriptions: United States subscriptions are $32.50 per year. Foreign subscriptions are $40 per year ($78 for Air Mail) and payable by Check on a US Bank or International Money Order *only.* For subscriber and advertising service please call (707) 887-7879.

Back Issues: Issues prior to April 1988 are available for $3 each ($8 with disk). Twelve issues of 1987 are available for $30.00. All prices postpaid.

ST Applications is in no way affiliated with Atari Corporation.

Atari, 520ST, 1040ST and the Atari logo are registered trademarks of Atari Corp.

GEM is a registered trademark of Digital Research Inc.

Views expressed herein are those of the authors and may not necessarily reflect the views of **ST Applications** or its staff. To the best of our knowledge the information contained herein is correct. However, **ST Applications,** its editors and staff cannot be held responsible for the use of the information or any damages which may result.

Second-Class Postage Rates paid at Forestville, CA 95436.

Postmaster: Send address changes to **ST Applications,** P.O. Box 980, Forestville, CA 95436.

Contents...

Vol. 4, No. 7 July 1988

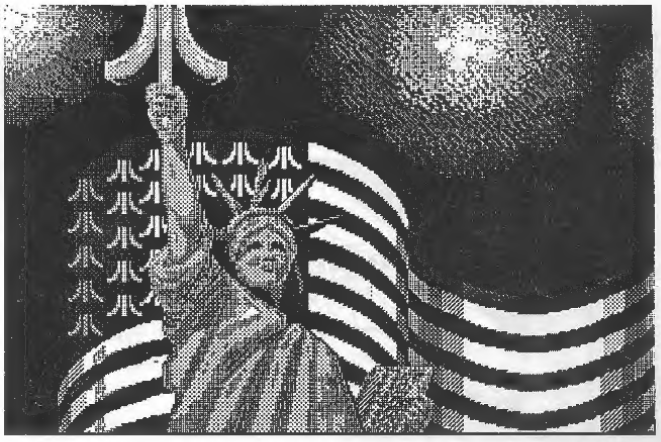
Departments

The Review Board

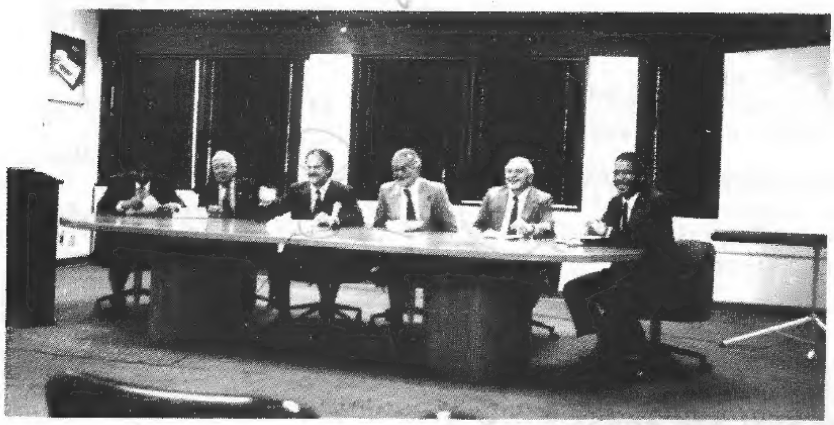
Indicates item(s) will be found on this month's Support Disk.

This Month's Cover:

The 3 color separation was made from public domain nofire1.neo with Eric Thornton's Platter2.prg at 150 dpi and 4 dots per pixel resolution. Thanks to the unknown artist.



Editorial



Atari Corp. Board of Directors

On Tuesday, 17 May 1988, at 2 pm, Atari Corp. held its second annual shareholder's meeting at Atari corporate headquarters, 1196 Borregas Avenue, Sunnyvale, California 94086.

Of the 57,725,242 shares outstanding, just over 49 million were represented by those in attendance or by proxy. The business conducted included election of board members for the coming year, financial report of the corporation, and a brief business outlook.

All members of last year's board were re-elected, and are listed as they appear in the photo from right to left.

Gregory A. Pratt
Vice President-Finance
Chief Financial Officer

Jack Tramiel
Chairman of the Board
Chief Executive Officer

Michael Rosenberg
Chairman and Chief Executive Officer-Ross & Roberts, Inc.

Sam Tramiel
President
Chief Operating Officer

Leonard I. Schreiber
Partner-Schreiber and McBride

Samuel W. L. Chin
Vice President

Printed by:

Robert John Industries

5351 Oceanus Drive
Huntington Beach, CA 92649

1988 First Quarter Results

Net Sales were $97,706,000 (without Federated) compared to $65,133,000 for the same period last year. Net income was $15,290,000 up from $15,258,000 last year.

The Federated Group reported net sales of $71,526,000 and a pretax operating loss of $9,616,000 for the first quarter of 1988. The losses were due to overhead (which had to be maintained throughout the Christmas selling season and has since been cut drastically) and creation of operating structures (supply channels, etc.) which were assumed to be part of the purchase but weren't. A modest profit is expected to be generated by this fall.

Brief Business Outlook

The non-ST, XE and game machines (2600 & 7800) are selling well, especially in Latin America and the Communist Block countries.

ST and related sales (about two thirds of total sales) are primarily in Europe. Due to the scarcity of **DRAM (Dynamic RAM)** chips which should be alleviated by late summer, PC production will be kept to a few thousand per month, and U.S. 'ST' sales will be light as Atari maintains its European market success.

Atari currently has no internal (ie. self manufactured) DRAM and will decide within 120 days of the shareholder's meeting whether it will start building a production facility or not. It will take approx. 1 year after making the decision to get such a facility in operation.

By late summer the impact of the restrictions on Japanese DRAMs should ease and at least this fall DRAMs can't be used as the scapegoat for limited production of memory using computers and peripherals.

Production of the Transputer is also scheduled for this fall (Sept, Oct), as is a full 68030 machine running Unix-V, the Atari CD ROM unit, Postscript driver for the Atari laser printer, and a 44 Mbyte, 20ms access time removable cartridge Hard Disk Drive for about $1000 (see photo below).

Mega ST with 44Mbyte cartridge hard drive

Food for Thought

On the buffet table at the back of the meeting room was a warming tray full of sliced hot dogs and next to it a small cup of caviar. On first impression this combination seemed quite odd. However, Jack Tramiel had his reasons. In business, you have to know when to pull in your belt and eat hot dogs, and when to splurge on caviar. He means that as shareholders now is the time to not expect dividends while Atari tightens its belt and gets its business in order. Once the business is running smoothly there'll be plenty of time (and money) for caviar. Of course, the first thing that popped into my head was an alternate explanation: that in business most of us eat hot dogs (i.e. 3rd party producers) and a veritable few at the top eat caviar.

Playing games at the shareholders meeting

Wrap Up

At last November's COMDEX Atari was proclaiming the ST line as business computers. As of this May's Shareholder's meeting Atari has still not given up the Personal Computer marketplace. Whether this is a one day change to the unpredictable no man's land of personal computing which is being abandoned by most other computer manufacturers or a definite trend is to be seen. Atari has also eliminated all distributors in the US and will have only area representatives, thereby selling directly to dealers.

Starting on the 21st of May, the Beverly Hills Federated store will begin selling ST computers (a full page ad in the Los Angeles Times kicked it off). If this strategy takes then we can look forward to 62 Federated ST outlets in the Southwest.

Lastly, Atari's newest image is one of innovator. They plan to offer leading edge technology at a low price no one can afford to pass up.

Thanks again,

Bill Petry

Bill Petry
Publisher

Letters

Dear ST,

ST Business is a great addition to your magazine! I'm not in business, but the info is useful, and I may make the step to small business some day.

I don't think there's a single article in ST Applications that I don't read and get something useful from. The basic columns, and the (late lamented) pascal column are the most useful, as are the languages I currently work with. LDW Basic and OSS Pascal specifically. Both are excellent for gem access, and general programing applications.

Again, thanks for the fine magazine.

Kris J. Gasteiger
Trumansburg, NY

Dear ST,

I would like to comment on the **Expert Opinion** program which was reviewed in the November, 87 issue. I am an amateur botanist interested in working out an easier way to identify wildflowers. The available books and keys are difficult to use and it's hard to identify a plant unless you already have a good idea of what it is.

When I read about **Expert Opinion** I thought this would make an excellent plant identification system. Rules can be set up to follow existing plant keys with the advantage that you don't have to follow the exact path that keys require. And other rules can be entered for unusual characteristics that never show up in plant keys.

So I bought the program and dove head first into **Expert Opinion** and came up gasping for air two days later. I would like to discuss what I consider to be two problems with the program. Before criticizing it I would like to say that there are many excellent features and I agree with most of what Donavan Vicha said in his review in November.

There are two different programs to EO. The Basemanager is the program that you use to write and edit rules for a knowledge base. The **Expert Opinion** program is the program with the inference engine used to consult the knowledge base. What is really needed is a shell program that would call these programs without having to return to the desk top. Having never written a knowledge base before, my way of doing it was to write a few rules then test them out, then write a few more and test again. I was constantly going back and forth between the two programs.

It is time consuming to exit a program to the desk top, wait for the disk directories to be displayed and read in a new program every few minutes. And when each program begins with a title screen that has to be clicked out of the way this becomes wearisome and irritating. But this is a problem that should be easy to rectify.

The other problem concerns the dictionary of words. Why? Why are the rules broken down into words? When entering facts to be scanned by the inference engine they have to be entered exactly the same way they are entered in the rules. Otherwise the program will not be able to recognize them. When building a fact from the words in the dictionary it is easy to make a fact not in the rules.

What is really needed is a list of the facts that make up the rules - not a list of the words that make up the facts. To give a specific exmple. From the animal knowledge base mentioned in the review, one of the rules is: IF the animal is a mammal AND the animal flies THEN the animal is a bat. From the dictionary it is easy to construct the following sentence: the animal flies in the forest. But the program is not able to use this sentence. It is a totally useless statement. I don't see any point in taking a sentence, breaking it up in little pieces and then requiring the user to reconstruct the statement in precisely the same way it was to begin with in order to use it. Why not just leave it alone?

In researching this problem I bought a book, "A Comprehensive Guide to AI and Expert Systems: Turbo Pascal Edition." In the chapter on forward chaining it presents a program which includes the following rules: IF interest = fall THEN stock = rise and IF interest = rise THEN stock = fall. In the program INTEREST and STOCK are considered to be variables and rise and fall are the values of the variables. The program is very complex and confusing and hard to follow even when you know what it's trying to do.

I am not an expert on expert systems and know very little about them and it's quite possible that I am missing an important fundamental point in expert system theory. If I am I would appreciate someone explaining it to me. What I really think is that the experts are taking something that ought to be simple and building a complex monstrosity out of it in order to derive a feeling of accomplishment. If it's not complicated it must not be expert.

I am solving the problem by writing my own version of an expert system using Personal Pascal version 2.01. It may turn out to be an inept system but I am off to a good start with it. I have a forward chaining program up and running and working well. I am keeping the rules in a list of records and assigning a number to each fact as it is entered in a rule. I multiply the string length by 100 and enter the fact and number in a list kept as a binary tree. If the fact matches one already in the list it receives the same number. During the chaining session the program does not have to search for strings - only integers. The forward chaining program begins by printing all the facts to the screen along with the fact numbers. To enter a fact, just enter the number from the keyboard.

For the backward chaining program a list of all the conclusions and numbers will be printed to the screen. When a conclusion is selected and a rule having that conclusion is found, all the facts in the rule will be printed to the screen. Then the screen will continue to seek other rules with the same conclusion.

To end this letter I would like to ask again if anyone can explain what the point is in breaking statements down into words in a dictionary that requires the user to reconstruct back to the original statement before they can be used? And also requires that the user do this blindfolded without knowing what the original statement was.

William Carswell
Ramona, CA

Dear ST,

I recently moved up from a 1040 to a Mega 4 system; 30 meg Hard Drive and all. The main reason I wanted more memory was the fact that I have been working on some extremely large programs. One function of these programs was to load document files from the disk drives into the program. Because of this I have had to write thousands of document files.

One evening while working with my work processor I had a problem when exiting back to the desk top. I found some of my files gone; including some of the documents I had just saved. In fact I found partition E of the hard drive completely empty. the first thing that came to mind was, "I lost it all." After sitting there in total shock for a few moments, knowing the only thing I could do was to shut the system down and reboot. With tears in my eyes I did just that.

Well, that's the sad side of the story; now for good news. After rebooting I found everything back to normal, so I quickly saved all my new files on drive E to some disks in drive A & B and put the aspirin bottle away.

This story does not end here, because this problem continued to haunt me; in fact, even after zeroing out partition E of the hard drive and restoring my files the problem was still there. By the way, I knew it wasn't my night when I even had trouble with this simple task. But that's another story.

This unsuccessfully didn't help one bit. The problem was still there. By now it was getting late and the night was well spent. I was starting to feel like Dr. Jeckle and Mr. Hyde; one minute happy the next minute sad. I continued to try different things, like renaming the ACC files and moving files out of the AUTO folder with still no avail.

As my eyes were getting heavy and I was just about to wrap it up, when, "BONK!" the light bulb lit! The thought ran through my mind; you dummy; you fool; you are overloading the desk top with too many files.

I quickly started looking through the Mega instruction manual to see how many files I could have on one desk top before the operating system could not read any more. But I had problems finding the answer. Well, I said to myself; "Self, why don't you just try overloading the desk top and see". So that's what I did.

I started by rebooting the system again, but this time I left the hard drive off and just opened with a disk in drive A. After opening the window and folder with 197 files on it I repeated it again. Now I had 394 files and two open windows on the monitor, and everything was still ok. I repeated this once more, but this time the open window only showed 6 files in the same folder when I knew there was 197 files. Great guns! I found the answer. The Mega is a wonderful and powerful machine, and it sure meets my needs, but I guess it does still have its limits; if you can call 400 files on one desk top a limit. By the way you can only have 4 open windows and then the system will warn you with an alert box if you try to open the fifth window. But as I found out, the system will not warn you when you try to open more than 400 files; it just refuses to read them.

I hope this little story will help you not make the same mistake I did.

Irving W. Risch
Wabasha, MN

Dear ST,

I wonder if you could help me solve a problem in C. It is related to ADCII characters and the manipulation of strings with the v_gtext function.

In BASIC one can declare the following:

```
spane$ = chr$(130)
spano$ = chr$(162)
symbol$ = chr$(189)
a$ = "C" + spano$ + "rdoba, Ver., M" + spane$ + "xico."
b$ = symbol$ + " Copyright."
print a$
print b$
```

In using the v_gtext function one must use the opcode 8 making a$ and b$ equal to strin$ or whatever parameter being considered (after having declared the proper parameters for the VDI environment like work_in, work_out, etc.).

The problem I have met in C is not relative to the TOS environment where it can be beautifully solved by the following implementation:

Continued on page 29 ==>

Picshell Menu Program

By Len Shikowitz

Generally speaking, the GEM Desktop is a wonderful metaphor. However, on those occasions that I sit down for "heavy computing sessions", it can be a less-than-efficient means for selecting the next program I want from whichever disk I want to run. During the course of the day I may be running GFA Basic, its compiler, an assembler, various paint programs, animation programs, etc. Selection of the program is inefficient with its limitation of listing 16 files per drive on screen in Text form (I hate selecting files from the Icon format, even though it permits up to 32 files on screen!), and the need to sort files by type in order to view only the MAJOR EXECUTABLE types (.PRG and .TOS) before selecting the program I want to run.

None of the multitude of "CLI" (Command Line Interface - MS-DOS-like programs) shells that I've seen are of any benefit in this regard either. First you would have to do a DIRectory call, either limiting it to .PRG or .TOS files, or showing ALL types of files. Then you have to type the name of the program that you wish to execute. This, too, was not representative of the type of efficiency that I was seeking.

So, left to my own devices, I created the "SHIK/SON PICSHELL Menu Program". Written in GFA Basic, this program will display any combination of up to 43 folders, .PRG and .TOS files per drive, on screen at one time. Selection of either the desired program or folder is made by pressing one key. An information line reflects the Path at all times. Selection of another drive (floppy, hard or RAM) is made by first pressing ESCAPE and then the letter of the drive. An information line tells you which drives are on line.

The program is structured as a "terminate and stay resident" program. That is to say, after you've executed the program of your choice (for argument's sake, "NEO") and quit that program, rather than returning to the GEM Desktop, you are returned to the "SHIK/SON Desktop", ready to select another program to be executed. As a bonus, you can choose to include (in the same directory as the PICSHELL program) a DEGAS screen file (.PI2 for color monitors and .PI3 for monochrome monitors). It will then be used as the background for the program.

Program Structure Outline

First, the program saves the present resolution and color pallette. If you are using a color monitor, the resolution is set to medium. The program then looks for a DEGAS file in the proper resolution. If one is present (in the same path as the program), it is loaded and used as the "Desktop" (I sometimes use a "snapshot of the GEM Desktop myself!). If none is present, the background and text colors are altered to black and yellow, respectively.

Next, the default drive is determined via a GEMDOS(25) call. The program (using a modified version

to exist), and executes the selected file. When the user exits the selected file, control is returned to the PIC-SHELL program together with all the memory. The user is then given the choice of either exiting to the "real" desktop or continuing to operate within the PIC-SHELL program.

I would like to thank the good people at Access to Software in San Francisco, who permitted me to test out this program on one of their systems which included two floppies, a hard drive partitioned into six logical drives, and a RAM disk. Since my personal system is a much simpler configuration, I would not have been able to confidently complete this program without their generous cooperation.

of the standard directory search and sort subroutines listed in the recently released "GFA BASIC BOOK" from Michtron) then finds, lists in alphabetical order, and in the three column format, all the folders, .PRG files and .TOS files (up to a useable combined maximum of 43) on the default drive. Folders are readily identified by a trailing "\".

The path is constantly monitored and modified based on user keypresses. Error trapping is provided to prevent invalid keypresses as well as selection of paths that contain neither folders nor executable (.PRG or .TOS) files. Once an executable file has been selected, the program simply frees up all unneeded memory via a RESERVE call (keeping enough for itself to continue

Picshell.Bas

```
' THE SHIK/SON MENU SHELL PROGRAM (Maximum 43 items)
Dim A$(50)
Dim B$(32)
Dim L$(50)
Dim Pal%(4)
Dim Pal2%(15)
Dim Sav%(15)
Dim Black%(15)
Rez=Xbios(4)
Pd%=0
If Rez=2 Then
  Pd$="BKGRND.PI3"
Else
  Pd$="BKGRND.PI2"
Endif
For J=0 To 15
  Zz=Xbios(7,J,-1)
  Sav%(J)=Zz
Next J
Close #1
If Exist(Pd$)=-1 Then
  Open "I",#1,Pd$
  For J=0 To 15
    Black%(J)=0
    C=Xbios(7,J,Black%)
  Next J
  @Degas
  For J=0 To 15
    Xx=Inp(#1)
    Yy=Inp(#1)
    Zzz=Xx*256+Yy
    Pal2%(J)=Zzz
  Next J
  Seek #1,Discard
  Bget #1,Xbios(2),32000
  Close #1
  Sget Screen$
  Cls
  Show:
  Close #1
  @Newpal
Else
  @Nextpal
  Pd%=1
Endif
Primary:
T%=Gemdos(25)+1
T$=Chr$(T%+64)+":"
M$="\"
Z$=Space$(14)
Zz$=Space$(3)
P%=0
Start:
If Rez=0 Then
  Void Xbios(5,L:-1,L:-1,W:1)
Endif
Cls
If Pd%=0 Then
  Sput Screen$
Endif
Print "                              SHIK/SON MENU PROGRAM"
Print
Print "    To select from a different disk, switch disks and then press RETURN
"
Print "                    FOLDERS are identified by a trailing '\'
"
C%=Bios(10)
Print "Drives on line are ";
B%=0
For I%=0 To Len(Bin$(C%))-1
  If Mid$(Bin$(C%),I%,1)="1" Then
    Print Chr$(65+I%);": ";
    B$(B%)=Chr$(65+I%)
    B%=B%+1
  Endif
Next I%
Print
Print T$+M$
Print
For I%=0 To N%
```

```
  A$(I%)=""
Next I%
@Getdir(M$+"*.*",&H37,*A$(),*N%)
If A$(0)="" Then
  Print "          There are no folders or executable files on this disk!
";Z$;
  Print "     Either replace disk and press RETURN, or select another
drive";Z$;
  Print "              by pressing ESCAPE and the letter of that drive.
";Z$;
  X=Inp(2)
  Goto Start2
Endif
@Quicksort(*A$(),0,N%)
If F%<>0 Then
  N%=N%-2
Else
  N%=N%+1
Endif
For Q%=0 To N%
  For I%=0 To N%
    If A$(I%)="" Then
      A$(I%)=A$(I%+1)
      A$(I%+1)=""
    Endif
  Next I%
Next Q%
If A$(I%)="" Then
  N%=N%-1
Endif
If A$(F%)="" Then
  If Pd%=0 Then
    $put Screen$
  Endif
  Print "  There are no additional folders or executable files in this folder.
"
  Pause 100
  F%=0
  T$=""
  M$=Chr$(T%+64)+":\"
  Goto Start
Endif
For I%=0 To N% Step 3
  If N%>25 Then
    Pr%=48+I%
    Pr2%=48
  Else
    Pr%=65+I%
    Pr2%=97
  Endif
  If A$(I%+2+F%)<>"" And N%<>0 Then
    Print Chr$(Pr%);") ";A$(I%+F%);Z$;Chr$(Pr%+1);")
";A$(I%+1+F%);Z$;Chr$(Pr%+2);") ";A$(I%+2+F%);Zz$
  Else
    If A$(I%+1+F%)<>"" Then
      Print Chr$(Pr%);") ";A$(I%+F%);Z$;Chr$(Pr%+1);") ";A$(I%+1+F%)
    Else
      Print Chr$(Pr%);") ";A$(I%+F%)
    Endif
  Endif
Next I%
Print
Print
Print
Print "Press the letter of your selection or ESCAPE+Drive Letter to change
drives    "
X=Inp(2)+F%
Start2:
If X=27+F% Then
  F%=0
  X=Inp(2)
  For I%=0 To B%-1
    If X=I%+97 Then
      T$=""
      M$=Chr$(X-32)+":\"
      Mm$=Left$(M$,2)
    Endif
  Next I%
  Goto Start
Endif
If X>96 And N%>25 Then
  X=X-32
  Pr2%=48
Endif
If X<Pr2%+F% Or X>Pr2%+N%+F% Then
  If F%=2 Then
    F%=0
    M$=Mm$+"\"
  Endif
  Goto Start
Endif
If Mid$(A$(X-Pr2%),13)="\" Then
  A$(X-Pr2%)=Left$(A$(X-Pr2%),12)
  F%=2
  For I%=0 To Len(A$(X-Pr2%))
    If Mid$(A$(X-Pr2%),I%,1)=" " Then
      A$(X-Pr2%)=Left$(A$(X-Pr2%),I%-1)
    Endif
  Next I%
  M$=M$+A$(X-Pr2%)+"\"
  Goto Start
Endif
@Original
If X>96 And N%>25 Then
  X=X-32
Endif
B%=Asc(Left$(M$,1))-64
Aa$=M$+Left$(A$(X-Pr2%),13)
Reserve 0
If B%<>28 Then
  Chdrive B%
Endif
Void Gemdos(59,L:Varptr(M$))
Exec 0,Aa$,"",""
Reserve Xbios(2)-Himem-16384+Fre(0)
If Pd%=0 Then
  @Newpal
Else
  @Nextpal
Endif
Cls
Void Xbios(5,L:-1,L:-1,W:1)
If Pd%=0 Then
  $put Screen$
Endif
Print "                                    SHIK/SON MENU PROGRAM"
Print
Print
Print
Print
Print
Print
Print " If you wish to exit to Desktop press RETURN. Press any other key to
continue. "
Desk%=Inp(2)
If Desk%<>13 Then
  If Pd%=0 Then
    @Newpal
  Else
    @Nextpal
  Endif
  Goto Primary
Endif
@Original
End
'
'
Procedure Getdir(File_$,Attr%,Str.arr%,Num.%)
  Local I_%,E_%,X$
  Swap *Str.arr%,File$()
  Void Gemdos(26,L:Basepage+128)
  File_$=File_$+Chr$(0)
  E_%=Gemdos(78,L:Varptr(File_$),Attr%)
  While E_%=0
    X$=Space$(13)
    Bmove Basepage+158,Varptr(X$),14
    X$=Left$(X$,Instr(X$,Chr$(0))-1)
    X$=Left$(X$+Space$(13),13)
    If Peek(Basepage+149) And 16
      L$(I%)="\"
```

Continued on page 50 ==>

Summer-Special! $30.00 for all 12 issues of 1987.

Single issues
$3 Each
($8 with disk)

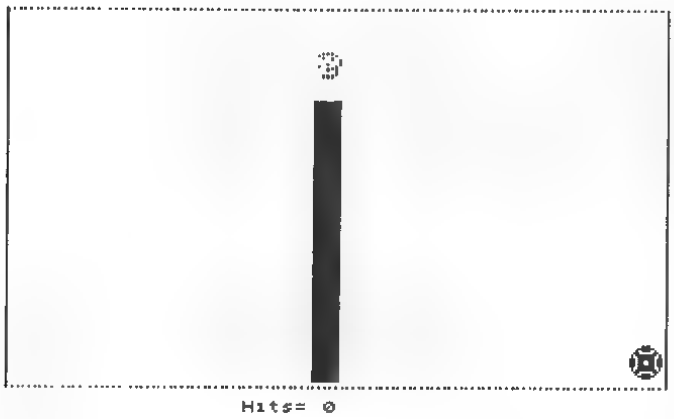
Vol. 1, two-disk set
$10.00 (text-files & programs
for 4 issues of 1985)

$27.50 for all 11 issues of 1986

Complete your set of ST Applications for only $3 each (less, when purchased in sets). Now, sets of the 12 issues for 1987 are available for $30. A limited quantity of complete sets for 1986 issues (11 total) are still available for only $27.50. Disk issues for all months prior to April 1987 are available for $5 ($8 with magazine). April, May and June1988 issues are available for $4.50 ($12 with disk).

Just send your Check of Money Order in the correct amount to:
ST Applications, P.O. Box 980, Forestville, CA 95436
< < < No dealer discounts may apply for this offer. > > >

The Wizard's Arena
An Add-On for 3D Maze Run



By James Dale Duncan

Here we go again. Another experiment in GFA Basic. This time we'll look at a short game program and see how we can use it to enhance the **Maze Run** program found in the May 1988 issue of **ST Applications.** I'm the first to admit that MAZERUN is really just a frame that needs much work. We could add sound or perhaps spruce up the titles. I prefer to work out the main game segments, merge and debug them, then put in the flicker and flash. With this in mind, I'd like to present **The Wizard's Arena.**

This program can be run as is, or with a few modifications, merged with **MAZERUN.BAS** to liven up that game. Their are three forms of the game on the support disk. **SKLSHOT.PRG** is the compiled version. (Just Double Click to run) **SKLSHOT.BAS** is the normal interpreter file and **SKLSHOT.LST** is an ASCII save that can be merged with MAZERUN (or any other GFA Basic program) using the interpreter. More on merging later. For now let's see what **SKLSHOT** does.

On running the program you get to see another one of my glorious (Ha!) title screens instructing you to hit five evil skulls or perish. Clicking the left mouse button clears the screen and puts your character at the bottom of the playfield. Moving the mouse right and left moves your character likewise. A grinning Skull will be bounding about the top of the screen, firing random bursts of Death Ray in your direction. Hit the left mouse button to return fire. This irritates the skull, causing him to move slightly closer to your level. These Skulls get **REAL** ticked off when hit and will make larger strides toward you.

Should the Skull reach your level, or hit you with a Death Ray, it's curtains. Hitting the required five Skulls ends the game rather abruptly, as does losing your one life. The death exit routine tells you what did you in, and reminds you that you had only one life, before exiting to the Desktop. Surviving the contest will cause the program to notify you of this and end the game.

This little game could be fleshed out a **LOT!** One place to start would be to allow for another play without returning to the Desktop. Since this program is intended to be added to **MAZERUN** it is brutally lacking in "*goodies*". I wanted to make grafting this listing to **MAZERUN** as simple as possible.

And it really is simple! Load up your GFA Basic interpreter **(GFABASIC.PRG)** and dig out your disk containing **MAZERUN.BAS** (this was published in the May, 1988 issue of ST Applications) and load that file. Move the cursor to the bottom of the listing. **<Control> + <Z>** will save having to listen to all those keyclicks by moving the cursor to the end of the listing in one move. Click on **MERGE** in the Menu bar. If **SKLSHOT.LST** is not on that disk, insert the disk that has it, and click on the little box to top left of the selec-

tor box to show those files. Double Click on **SKLSHOT.LST** and it will miraculously appear beginning at the current cursor position. (I used to program the Tandy Color Computer and to me it **IS** miraculous!)

We still have a bit more work to do before we're done. There are five lines boxed by asterisks (*) at the top of the listing we just merged. Position the cursor at the first of these (one line below the asterisks that form the top of the box) and click **BLOCK START** in the Menu bar. Move the cursor to the line that forms the bottom of the box and click on **BLOCK END**. Page up through the listing with the <Control> <Up Arrow> until you find Procedure Getkey. Position the cursor one line below this and click on BLOCK. Click on MOVE and the five selected lines are move to their new home.

Before moving on delete the asterisks and the apostrophes in each of these lines. Now **<Control> + <Z>** to the end of the file and page back up, scanning for **Procedure Hit**. In this procedure there are four lines with apostrophes as the first character. Delete the apostrophes and change **"End"** to **"Gosub Getkey"** as noted in the listing. That's it! Believe me, it is much quicker and easier to do than it is to write out how it's done!

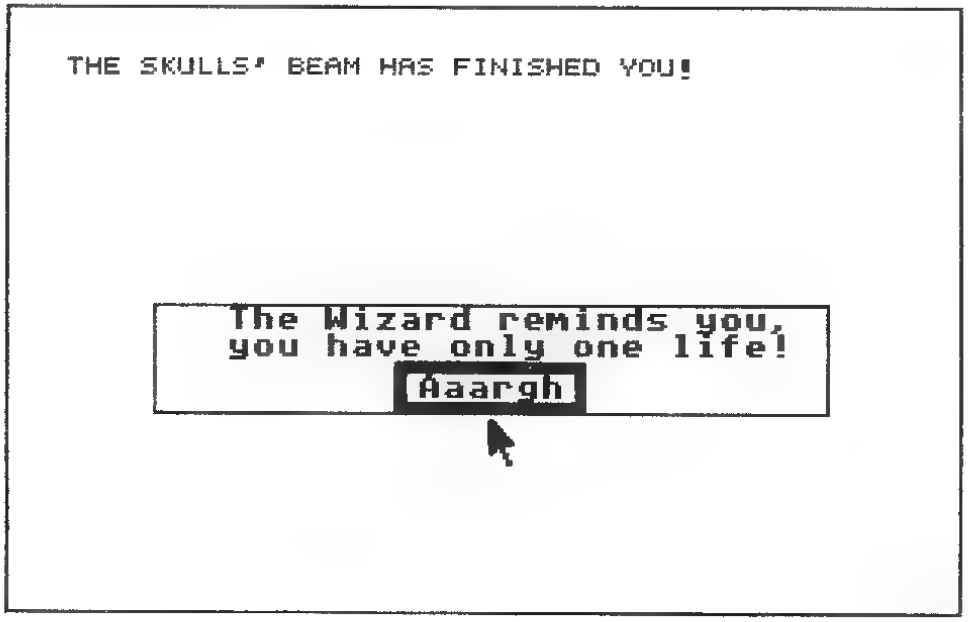
Save the combined program to disk and test run it. You will need a maze data file for **MAZERUN**. Everything works normally until ... **ZAP!** , you find yourself confronted with the intro page for the **Wizard's Arena**. Should you survive, you will be zipped back to the maze room you exited from. In the event of your demise, the entire game ends. Play the game a few times, make some notes, and rewrite an ending to suit youself. The exit to the **Wizard's Arena** is presently keyed to the height of the maze array. In other words, if the maze is five rooms high the player will wind up in the Arena every five moves. These are both points that could use much improvment.

There are two points about the program that may be of interest. The first is the use of **Sget** and **Sput**. When exiting the maze either for help or to the Arena, the current screen is stored in the variable **Room$** using the command: **Sget Room$**. In returning to the maze display we use the command **Sput Room$** to restore the maze screen. Once back in the maze we free up the memory used by **Room$** with: **Room$=""**. These are very fast commands with tons of possibilities.

You may note the use of the % (percent) symbol in the **SKLSHOT.BAS** variables while it is absent in the **MAZERUN** program. The use of the % symbol makes the variable an integer type variable. This cuts down on processor time when you don't really need 11 digit accuracy. A side effect is that it also makes X% totally unique from X. A cheap and dirty trick to be sure. But it can help prevent the programmer from declaring the same variable twice when writing a quick merge file at 3 AM!

Well, that ought to have you GFA Basic users loaded with ideas for awhile. As always, your comments and suggestions are welcome and encouraged! If any of you have any neat hacks of **MAZERUN**, let's see them in **ST Applications**. For now, Catch you on the Bitstream!

MORE



```
'
' *************** skulshot.bas **************
' Add-on for 3D Maze Run by James Dale Duncan
'  For ST Applications in GFA Basic
'

'
' The program will run stand-alone as written.
' Add the 5 lines below to the beginning of
  "Procedure Getkey" in the program 3DMAZERUN
  and make the noted edits to merge the 2 programs
  into one continous game.
  See article in ST Applications.

  ****************************************
' * Inc Shootemup                        *
' * If Shootemup=H                       *
  * Sget Room$                           *
' * Gosub Readsprite                     *
' * Endif                                *
' ****************************************
'
'
Gosub Readsprite
'
'
'
Procedure Readsprite
  ' Sprite-Convert data in string
  Let Blkboppr$=Mki$(7)+Mki$(8)
  Let Blkboppr$=Blkboppr$+Mki$(0)
  Let Blkboppr$=Blkboppr$+Mki$(0)
  Let Blkboppr$=Blkboppr$+Mki$(1)
  For I%=1 To 16
    Read Foregrnd,Backgrnd
    Let Blkboppr$=Blkboppr$+Mki$(Backgrnd)+Mki$(Foregrnd)
  Next I%
  Data 960,0,4080,0,5064,0,14364,0,24570,0,20466,0,61431,0,61047,0
  Data 61047,0,61431,0,20466,0,24570,0,14364,0,5064,0,4080,0,960,0
  '
  '
  ' Sprite-Convert data in string
  Let Skull$=Mki$(7)+Mki$(7)
  Let Skull$=Skull$+Mki$(0)
  Let Skull$=Skull$+Mki$(4)
  Let Skull$=Skull$+Mki$(2)
  For I%=1 To 16
    Read Foregrnd,Backgrnd
    Let Skull$=Skull$+Mki$(Backgrnd)+Mki$(Foregrnd)
  Next I%
  Data 0,0,4064,0,4112,0,11464,0,10380,1088,8716,0,8972,0,4120,0
  Data 2360,1728,2736,256,2096,0,2032,0,2016,0,0,0,0,0,0,0
  '
  '
  '
  Restore
  Gosub Title2
  Gosub Newgame
  Gosub Game
  '
  '
Return
'
'
Procedure Title2
  Cls
  Deftext 1,1,0,4
  Text 95,25,' THE WIZARDS' ARENA"
  Text 95,45,"Copyright  (c) 9/1987'
  Text 115,65,"By Dale Duncan'
  Text 98,55,"Written in GFA Basic"
  Text 10,150,"Use mouse to move. Left button fires Skull-Splitter."
  Text 10,160,"Hit the bounding Skull 5 times to survive."
  Text 10,170,"If Skull shoots or reaches you, ADIOS, Muchachos."
  Text 10,180,"Press right button to exit game COMPLETELY!"
  A$="ST Applications  |P.O. Box 980  |Forestville,CA  |(707) 887-7879  "
  Alert 0,A$,1,"BEGIN",A1
  If A1=1
    Gosub Newgame
  Endif
Return
'
Procedure Drawscreen
  Cls
  Bx%=50
  By%=75
  Ybottom%=50
  X%=150
  Y%=120
  Defline 1,1,0,0
  Color 4
  Draw 0,0 To 310,0 To 310,179 To 0,179 To 0,0
  Deffill 0,1
  Defline 1,12,0,0
  Deftext 1,0,0,4
  Text 100,190,"  Hits="
  Text 150,190,Sc%
Return
'
Procedure Newgame
  Lives%=1
  H%=1
  My%=100
  Y%=167
  Ybottom%=50
  V%=1
  Z%=1
  Sc%=0
  Gosub Drawscreen
  Hiden
Return
'
Procedure Game
  Repeat
    Gosub Bopmov
    Blaser%=Random(100)
    If Blaser%=5
      Gosub Blaserdown
    Endif
    Gosub Skullmov
  Until K%=2
Return
'
Procedure Bopmov
  Mouse Mx%,My%,K%
  If Mx%>190
    Add X%,2
  Endif
  If Mx%<110
    Add X%,-2
  Endif
  If X%>300
    X%=300
  Endif
  If X%<15
    X%=15
  Endif
  Y%=167
  Sprite Blkboppr$,X%,Y%
  If Point(X%,Y%)<>0
    Gosub Dead
  Endif
  If K%=1
    Gosub Laserup
  Endif
Return
'
Procedure Skullmov
  Add Bx%,H%
  Add By%,V%
  If Bx%>=300
    H%=-Z%
  Endif
  If Bx%<=15
    H%=Z%
  Endif
  If By%>=Ybottom%
    V%=-Z%
  Endif
  If By%<=15
```

```
    V%=Z%
  Endif
  Sprite Skull$,Bx%,By%
Return
'
Procedure Dead
  Lives%=Lives%-1
  Color 0
  R%=4
  Repeat
    Circle X%,Y%,R%
    Dec R%
    Sound 1,15,R%,4,2
  Until R%=0
  Sound 1,0,0,0
  Bx%=50
  By%=75
  X%=15
  Y%=12
  Sprite Blkboppr$,X%,Y%
  Gosub Drawscreen
  If Lives%<=0
    Gosub Die
  Else
    Gosub Game
  Endif
Return
'
Procedure Hit
  If Point(Bx%,By%)<>0
    Add Ybottom%,10
    If Ybottom%>=140
      Gosub Die
    Endif
    Bx%=Random(300)+10
    Add Z%,1
    Sc%=Sc%+1
    Text 150,190,Sc%
    If Sc%=5
      Cls
      Text 125,100,"YOU SURVIVED! "
      Pause 200
      Defline 1,1,0,0
      '       Shootemup=0  ! Remove apostrophe if used with MAZERUN
      '       Sput Room$   ! Remove apostrophe if used with MAZERUN
      '       Room$="M"    ! Remove apostrophe if used with MAZERUN
      End                  ! Replace with "Gosub Getkey" if used with MAZERUN
    Endif
    Color 0
    R%=4
    Repeat
      Circle Bx%,By%,R%
      Dec R%
      Sound 1,15,R%,7,2
    Until R%=0
    Sound 1,0,0,0
  Endif
Return
'
Procedure Zap1
  For Zap=1 To 4
    Sound 2,15,Zap,8,1
  Next Zap
  Sound 2,0,0,0
Return
'
Procedure Zap2
  For Zap=1 To 4
    Sound 2,15,Zap,6,1
  Next Zap
  Sound 2,0,0,0
Return
'
Procedure Laserup
  Color 1
  Line X%,Y%-10,X%,2
  Gosub Zap1
  Gosub Hit
  Color 0
  Line X%,Y%-10,X%,2
  Color 1
  Inc Ybottom%
  If Ybottom%>=140
    Gosub Die
  Endif
Return
'
'
Procedure Blaserdown
  Color 1
  Line Bx%,By%+8,Bx%,177
  Gosub Zap2
  Bopzap%=Point(X%,Y%)
  Color 0
  Line Bx%,By%+8,Bx%,177
  Color 1
  If Bopzap%<>0
    Gosub Dead
  Endif
Return
'
'
Procedure Die
  If Ybottom%>=140
    Sprite Skull$
    Sprite Blkboppr$
    Text 20,20,"THE SKULL HAS REACHED AND DEVOURED YOU."
  Endif
  If Lives%=<0
    Text 20,20,"THE SKULLS' BEAM HAS FINISHED YOU!"
  Endif
  Repeat
    Mouse X,Y,K
  Until K=0
  A$="The Wizard reminds you,   !you have only one life!   "
  Alert 0,A$,1,"Aaargh",A1
  If A1=1
    End
  Endif
Return                                                    END
```

An Account of the Creation of "STocks and Bonds"

by Anthony L. Farmer

Introduction

It all started with my strong desire to play Avalon Hill's board game "**Stocks and Bonds.**" The problem I encountered was that no one around me seemed to want to play the game very much. Since the game is fairly boring when played solo, I turned to my 1040-ST for an answer. Search as I might, I could not find a stock market game or simulation for the ST, so I set out to write my own version of "**Stocks and Bonds.**" I decided that the project would provide me with a good chance to learn about programming for GEM in Personal PASCAL, and be a lot of fun to boot. It would also provide a perfect opportunity to show that **anybody** can make a program which works on both color and monochrome ST systems. Up until the release of my "**STocks & Bonds**" program, most monochrome ST owners were left out in the cold when it came to games.

As the project progressed, I decided to really spruce it up and try to market the finished product with Avalon Hill's computer games division. The response from Avalon Hill was pathetic and depressing, however. Their bottom line was essentially that producing for the Atari ST market was financially unsound and that they had no plans to ever release "**Stocks and Bonds**" for the ST. With that understanding, I asked for and received verbal and written consent from them to release my "**STocks & Bonds**" game into the Atari ST public domain. Thus, version 1.0 of the program was born and posted on GEnie.

Version 1.0 proved to be quite popular and received kind and charitable comments from many users, but (like most software) suffered from several bugs and "dysfunctional features." Version 2.0 fixed every bug in "**STocks & Bonds**" that I was consciously aware of and added several nice new features. For the aesthetic values, I added a digitized picture for a title screen. My thanks for the picture's raw digitization still goes out to Fred Beckman, former GEnie SysOp.

About a month ago (early May, 1988), I submitted a review of "Stock Market: The Game" to **ST-Applications** (June issue) magazine along with version 2.0 of "**STocks & Bonds**" for distribution on their monthly disk. At that time, they also agreed to accept the article you are now reading. I wanted to give **ST-Applications** readers a bonus with this article, so I decided to submit "**STocks & Bonds**" 3.0, which you should find on the disk accompanying this issue. Since there weren't really any bugs (that I knew of) to fix, I had to add some features for version three. And boy did I add them! The latest and greatest version of "**STocks & Bonds**" contains several enhancements, but the most notable is the addition of **digitized sound.** I used MicroDeal's "ST-Replay" utility to digitize the sounds and linked in Tim Purves' "replay" routine to play the sounds back. Within the source code

for "STocks & Bonds" 3.0, you will find one of the very few examples of how to playback ST-Replay sound files from within Personal PASCAL.

Technical Background Information

Most of what follows in this article is a highest-level explanation of how the "STocks & Bonds" program does what it does. For those interested in the actual Personal PASCAL implementation of the program and its functions, a WELL-documented source code is provided at the end of this article and on the **ST-Applications** disk for this month.

From the outset of this project, very modular code and top-down design have been a self-imposed requirement. The ease of maintainability and readability has been well worth the cost of my original decision. As you can tell from the program listing, modularity has made the main program module short and readable by even the most casual observer (grin).

Because many of the Atari OS and GEM calls are not built into Personal PASCAL (P.P. from here on), and because P.P. (like most PASCAL compilers) lacks a random number generator, I had to write several short utility routines for use by "STocks & Bonds." You may find these routines useful in your own programs.

Data File Formats

Version 3.0 of "STocks & Bonds" has a grand total of 10 data files that fall into 3 categories: ST-Replay sound files, DEGAS picture files, and custom text data files. As stated, the sound (.SPL) files are in standard ST-Replay format and can be fiddled with to your hearts content. Just remember to play with **backups!** The picture files (.PIC) are in NON-compressed DEGAS format. The MONO.PIC file is a high-resolution image, and the COLOR.PIC file is a medium-resolution picture. Obviously, these too may be backed-up and then fiddled with at your leisure. The three .DAT files are text files that contain information on the various stocks in the game. The exact formats for these files are described below. You can use almost any **TEXT EDITOR** to modify the .DAT files. You can also use a word processor that allows you to "print to disk."

Format of MARKET.DAT

This file consists of two sequential 9-row by 11-column tables. The first table is used in determining stock price fluctuations during a bear market year and the second table is used for doing the same during a bull market year. The format of the two tables is identical: The 9 rows of a table represent the stocks in the game and the 11 columns represent the possible values of two 6-sided dice. For example, if year two is a bull market, the second table is consulted by "rolling two 6-sided dice" and reading down the column indicated by the dice, for each stock. The values in the table represent how much each stock goes up or down for a given year. Note that there is no row for bonds! Bond prices do not fluctuate in this game!

Format of SBINFO.DAT

This file consists of ten 10-line records, one record for each stock and one for the bond. The first line in a record is a 30-character string that contains the **full** name of the stock/bond. The second line is a 13-character string containing the **short** name of the stock/bond. This **short** name is used to represent the stock in dialog boxes. The third line of a record is an integer that represents the yield of the stock/bond. The integer is a percentage (without the % sign). The next seven lines are 80-character strings that, together, make up what is displayed when the user asks for info on that stock/bond. These last seven lines must be manually centered when modified with a text editor. Note that if you do not have 7 lines of text for a given stock description, you **must** leave enough blank lines to make a total of 7 lines in the description.

Format of SBCARDS.DAT

This last file consists of thirty-six 9-line records. These records make up the deck of "cards" that are used to induce random stock price fluctuations during the course of a game. Each year (turn) one "card" is drawn. This card determines whether the year is a bull or a bear market year and randomly affects from 1 to 4 stocks. The first line of each record is an integer that can be equal to 0 or 1, making the turn that the card was drawn in, either a bear (0) or bull (1) market. The next 4 lines contain 80-character strings which, together, make up what is displayed as a "NEWS FLASH" when that card is drawn. When edited, these 4 lines must be manually centered and there must be **exactly** 4 lines of text, so enter blank lines to make up the difference if you don't have 4 lines! Each of the last four lines in a record contains a pair of integers. The first in a pair is the number of a stock affected (as defined in SBINFO.DAT) and the second integer in the pair is the amount of price fluctuation (positive or negative) to add to the number gotten from the MARKET tables for the turn in which the card was drawn. If less than 4 stocks are affected, use a pair of zeroes to make up the other pairs. Note that the price fluctuation information in the "NEWS FLASH" corresponds **directly** with the 4 pairs of integers!

Modules used in STocks & Bonds

The following FUNCTIONS were written for use by STocks & Bonds:

GET_REZ : An XBIOS call to determine screen resolution.

RND : An XBIOS call that generates a 24-bit random number.

RAND : My own random number (between 0 and 1) generator.

Stock	Yr1	Yr2	Yr3	Yr4	Yr5	Yr6	Yr7	Yr8	Yr9	Y10
GROWTH CORPORATION OF AMERICA	123									
METRO PROPERTIES, INC.	128									
PIONEER MUTUAL FUND	98									
SHADY BROOKS DEVELOPMENT	111									
STRYKER DRILLING COMPANY	78									
TRI-CITY TRANSPORT COMPANY	119									
UNITED AUTO COMPANY	130									
URANIUM ENTERPRISES, INC.	129									
VALLEY POWER AND LIGHT COMPANY	114									

The following is a list of PROCEDURES called by **"STocks & Bonds":**

REPLAY: P.D. routine supplied by MichTron. Used to replay digitized sounds. No source code was provided. The object file (supplied on this month's **ST-Applications** disk) is in standard DRI format.

CONT_MENU: A simple routine that waits for any keyboard event or the right-mouse-button-pressed event and then returns.

SAY_HELLO: Displays the copyright notice.

END_GAME: Determines the final scores of all players and displays player rankings.

INIT_VARS: Initializes all things that only need it once, such as the reading in of the data files and determining screen resolution.

INIT_AGAIN: Initializes variables that need initializing at the beginning of every game, such as the first year's stock prices, shuffling cards, etc.

INIT_MENUS: Initializes all of the dialog boxes used.

MARKET: Displays the stock board.

UPDATE_PLAYER: Gives players their due dividends each turn.

UPDATE_MARKET: Draw a card, display NEWS FLASHES, determine the new year's stock prices, handle stock splits and worthless stock in each player's portfolio. Called every turn.

SEE_PORTFOLIO: Display the current player's portfolio.

GET_INFO: Display information on a selected stock.

BUY_STOCK: Allow current player to purchase stocks.

SELL_STOCK: Allow current player to sell stocks.

SEE_MONO: Display graph of selected stock in monochrome.

SEE_COLOR: Display graph of selected stock in color.

Program Flow

In the "big picture," it is important to know that all major variables are kept global because most of the routines in **"STocks & Bonds"** access many of the variables. Having the variables global avoids unnecessary complications with parameter passing. It is also important for you to know that many compromises were made in the user interface because of bugs in Personal PASCAL. One of the biggest compromises came about when I wanted to display the stock board and the main menu on the same screen. I wanted to do so because the "SEE STOCK BOARD" main menu option is used the most. I could not do it, however, because the current version of Personal PASCAL will not display a dialog box that is not centered. This **MAJOR** bug has caused many people some discompfort. Here, is the top-level procedure flow for **"STocks & Bonds:"**

```
 * Set the mouse cursor to a pointing hand
 * Call INIT_VARS
 * Call INIT_MENUS
 * Call SAY_HELLO
 * Loop until the user enters zero for the number of
players
 - Call INIT_AGAIN
 - Enter the number of players
 - Enter the names of all players
 - Loop 10 times or until a user wishes to quit
 : Display and do the main menu
    + If "NEXT TURN" chosen
 = Call UPDATE_PLAYER
 = Call UPDATE_MARKET
 + Call the routine appropriate to the user's choice
 - Call END_GAME
 * Delete all dialog boxes to free up memory
 * Reset Mouse Pointer
 * Exit to the Desktop
```

Conclusion

This wraps up my top-level discussion of what makes **"STocks & Bonds"** tick. With this discussion and the source code in hand, you should be ready to have all kinds of fun customizing the program. I certainly hope you have enjoyed this article and my **"STocks & Bonds"** program. If you have any comments or suggestions, please send them to me at ALF ENGINEERING / P.O. Box 2964 / Rapid City, SD 57709. You can also find me hanging around on GEnie and my username there is, of course, ALF. If you would like to see more articles like this in **ST-Applications**, please write and let them know. Both they and I would appreciate it.

```
{$L120}          { Need this much heap space for digitized sound }

PROGRAM Stocks_and_Bonds ;

{ This program is the Atari ST version of the Avalon Hill board game }
{ of the same name.  It will work on any Atari ST (as far as I know) }
{ and requires OSS's Personal PASCAL 2 xx to compile.  Also needed   }
{ is MichTron's REPLAY.O object file.  Link this file in so that the }
{ REPLAY routine can be called to replay digitized sounds.  Other    }
{ modules called include:  GET_REZ, RND, RAND, CONT_MENU, MARKET,     }
{ UPDATE_PLAYER, UPDATE_MARKET, SEE_PORTFOLIO, SEE_MEMO, SEE_COLOR   }
{ GET_INFO, BUY_STOCK, SELL_STOCK, SAY_HELLO, END_GAME, INIT_AGAIN,  }
{ INIT_VARS, and INIT_MENUS.  This source code and all subroutines   }
{ except REPLAY.O are copyright 1988 by Anthony L. Farmer and ALF    }
{ Engineering.  Current version of software is 3.0 / 05-20-88        }

{$I D:\GEMSUBS.PAS}
{$I D:\AUXSUBS.PAS}

CONST
  MIN = 0;          { Stocks are numbered from 0 to 9 as read from }
  MAX = 9;          { SBINFO.DAT }

  VIDEO = $1180;    { Video options for displaying menus }
  MY_RED = $1280;
  MY_GREEN = $1380;

  SOPEN = 0 ;       { Index to snd buffer for opening music }
  SBYE  = 0 ;       { Index to snd_buffer for GOODBYE sound }
  SBULL = 18760 ;   { Index to snd_buffer for BULL MARKET sound }
  SBEAR = 27990 ;   { Index to snd_buffer for BEAR MARKET sound }

  LOPEN = 87062;    { Length of opening music (in bytes) }
  LBYE  = 37512;    { Length of GOODBYE sound (in bytes) }
  LBULL = 16427;    { Length of BULL MARKET sound (in bytes) }
  LBEAR = 28850;    { Length of BEAR MARKET sound (in bytes) }
  LPRINT = 6326;    { Length of PRINTER sound (in bytes) }

  KHZ10 = 2 ;       { 10 Khz playback }
  KHZ15 = 3 ;       { 15 Khz playback }
  OPSIZ = 43540;    { Size of snd_buffer array = LOPEN/2 }
  PRTBUF = 3164;    { Size of print_snd array }

TYPE
  Ctype = RECORD      { Card type as read from SBCARDS.DAT }
    bulbar   : Integer;      { Bull or Bear }
    errata   : Array [ 0..3 ] Of String[ 80 ];     { Stock market news }
    stocknum : Array [ 0..3 ] Of Integer;
        { Number of the stock affected by news }
        { Ignore any zero entry here }
    amount   : Array [ 0..3 ] Of Integer;
        { The amount ( + or - ) stocknum was affected }
        { Ignore any zero entry here }
  END;

  { INFO Type -- Stock descriptions from SBINFO.DAT }
  Itype = Array [ MIN..MAX, 0..6 ] Of String[ 80 ];

  { Market Type -- As per tables in MARKET.DAT }
  Mtype = Array [ 0..1, 1..9, 1..11 ] Of Integer;

  { Stock Type }
  Stype = RECORD
    name  : String[ 30 ];    { Name of the stock }
    short : String[ 13 ];    { Short name used in dialog boxes }
    price : Array [ MIN..MAX ] Of Integer;     { Price for each year }
    yield : Integer;         { yield * shares = money gained that year }
    split : Array [ MIN..MAX ] Of Boolean;   { TRUE = stock split THIS year }
  END;

  { Player Type }
  Ptype = RECORD
    pname : String[ 255 ];  { Player's name }
    { Amount of each stock owned }
    stock_owned : Array [ MIN..MAX ] Of Long_Integer;
    { Max price each owned stock was purchased at }
    bought : Array [ MIN..MAX ] Of Long_Integer;
    balance     : Long_Integer;  { Cash on hand }
  END;

  { Type for digitized sound files }
  SPL type = FILE OF Integer;

VAR
  nam_menu : Dialog_Ptr;
  nam_val,
  nam_done
  nam_opt,
  nam_prompt : Integer,

  bond_menu : Dialog_Ptr;
  b_val,
  bond_opt : Integer;
  b_typ : Array [ 0..3 ] Of Integer;

  buy_menu : Dialog_Ptr;
  pitch,
  buy_opt : Integer;
  pick : Array [ 0..10 ] Of Integer;

  start_menu : Dialog_Ptr;
  start_opt,
  one,
  two,
  three,
  four,
  five,
  six,
  seven,
  eight,
  nun : Integer;

  main_menu : Dialog_Ptr;
  main_opt,
  text1,
  text2,
  board,
  next,
  pfolio,
  quit,
  graphs,
  buy,
  sell,
  sb_info : Integer ;

  stock_menu : Dialog_Ptr;
  stock_opt,
  bond,
  growth,
  metro,
  pioneer,
  uranium,
  shady,
  stryker,
  tran,
  auto,
  power : Integer;

  cards      : Array [ 0..35 ] Of Ctype;   { Actual cards data }
  card_table : Array [ 0..35 ] Of Integer; { Card index table used in }
                                           { shuffling cards }
  bear_bull  : Ntype;    { 0 = Bear Mkt, 1 = Bull Mkt }
  info       : Itype;    { Info on the stocks }
  player     : Array[ 1..8 ] Of Ptype;
  stocks     : Array [ MIN..MAX ] Of Stype;

  rez_flag,                { FALSE = Medium rez, TRUE = High rez }
  skip_flag,               { Set to skip the CONTINUE dialog box }
  quit_flag2,              { Set to totally exit the program }
  quit_flag : Boolean;       Set to quit an individual game }

  { Misc. variables used in tabular data output, such as the portfolio }
  port_line,
  port_header,
  header,
  myline   : String[ 80 ],
  tmpstr   : String[ 255 ];  { Used in GEM calls requiring a STRING[255] }

  seed,                { Random number generator seed - MUST BE GLOBAL! }
```

MORE...

```
  temp,               { Used for whatever tickles your pickle }
  players,            { Number of players }
  who,                { Player currently up to bat }
  year : Integer;     { Current year }

  pic_file   : String[ 80 ];      { Filename of title picture }
  screen_buf : Screen_Type;       { Buffer for title screen }

  print_ptr,   { Pointer to print_snd buffer }
  bye_ptr,     { These 3 are pointers to SPL data within snd_buffer }
  bull_ptr,    { Note that bye_ptr is also used for opening music }
  bear_ptr   : Long_Integer;

  { Buffer for all sound files except the printer sound }
  snd_buffer : Array[ 0 .. OPSIZ ] Of Integer;

  { Buffer for the printer sound -- added after the others }
  print_snd  : Array[ 0 .. PRTBUF ] Of Integer;
PROCEDURE Replay( data_ptr, lenght, freq: Long_Integer );
  C;
{ Declaration for MichTron's replay routine gotten from REPLAY.O
   data_ptr : Pointer to array of integers
     lenght : Length, in bytes, of the raw SPL data.  SPL data is an array
              of integers with actual data only in the low-order bytes
       freq : Frequency for replay speed from p. 25 in manual;  0 = 5Khz,
              1 = 7.5Khz, 2 = 10Khz, 3 = 15Khz, 4 = 20Khz, 5 = 31Khz }


{$I A:\SB_UTILS.INC}

{$I A:\SB_SUBS1.INC}

{$I A:\SB_GRAPH.INC}

{$I A:\SB_SUBS2.INC}

{$I A:\SB_INIT.INC}


{ ********************  MAIN PROGRAM BEGINS HERE  ********************** }

BEGIN
  IF (Init_Gem >= 0) THEN
    BEGIN
      Clear_Screen;
      Init_Mouse;
      Hide_Mouse;
      Clear_Screen;
      Curs_Home;
      Set_Mouse ( M_Point_Hand );

      Writeln;
      Writeln('Creating stock market.  Please wait...');
      Writeln;

      Init_Vars;
      Init_Menus;

      { Set colors again for Medium Rez because they were screwed up }
      { by the loading of the title screen }
      IF ( rez_flag = FALSE ) THEN
        BEGIN
          Set_Color( 2, 0, 1000, 0 );     { Set RED register to green color }
          Set_Color( 3, 0, 0, 1000 );     { Set GREEN register to blue color }
        END;

      { Display copyright notice }
      Say_Hello;

      quit_flag2 := FALSE;

      WHILE ( quit_flag2 = FALSE ) DO
      BEGIN
        { Re-Init stuff for a new game }
        Init_Again;
```

```
{ Determine the number of players }
Show_Mouse;
Clear_Screen;
start_opt := Do_Dialog( start_menu, 0 );
End_Dialog( start_menu );
Clear_Screen;
Hide_Mouse;
Clear_Screen;
Curs_Home;
{ Reset chosen object }
Obj_SetState( start_menu, start_opt, Normal, FALSE );

{ Sorry about the sloppy IF-THEN structure below, but stupid ole' }
{ PASCAL will only let you use constants for CASES. ACK! PTHPT! }

IF (start_opt = one) THEN
  players := 1
ELSE
  IF (start_opt = two) THEN
    players := 2
  ELSE
    IF (start_opt = three) THEN
      players := 3
    ELSE
      IF (start_opt = four) THEN
        players := 4
      ELSE
        IF (start_opt = five) THEN
          players := 5
        ELSE
          IF (start_opt = six) THEN
            players := 6
          ELSE
            IF (start_opt = seven) THEN
              players := 7
            ELSE
              IF (start_opt = eight) THEN
                players := 8
              ELSE
                IF (start_opt = nun) THEN
                  BEGIN
                    quit_flag2 := TRUE;
                    quit_flag := TRUE;
                  END;


{ Have players enter their names }
IF ( quit_flag2 = FALSE ) THEN
  FOR temp := 1 TO players DO
  BEGIN
    Show_Mouse;
    Clear_Screen;
    { Change prompt for each player }
    WriteV( tmpstr, 'Name of Player ', temp:1, '?' );
    Set_DText( nam_menu,nam_prompt,tmpstr,System_Font,TE_Center );
    nam_val := Do_Dialog( nam_menu, nam_opt );
    End_Dialog( nam_menu );
    Clear_Screen;
    Hide_Mouse;
    Clear_Screen;
    { Get player's name from dialog tree }
    Get_DEdit( nam_menu, nam_opt, player[ temp ].pname );
    { Reset chosen objects }
    Set_DText( nam_menu, nam_opt, '', System_Font, TE_Center );
    Obj_SetState( nam_menu, nam_val, Normal, FALSE );
  END;


While ( (quit_flag = FALSE) AND (year <= MAX) ) DO
BEGIN
  { Indicate whos turn it is }
  Set_DText( main_menu, text2, player[ who ].pname,
        System_Font, TE_Center);

  { Let player choose what to do - main menu }
  Show_Mouse;
  Clear_Screen;
  main_opt  := Do_Dialog( main_menu, 0 );
  End_Dialog( main_menu );
```

```
Clear_Screen;
Hide_Mouse;
Clear_Screen;
Curs_Home;
{ Reset chosen object }
Obj_SetState( main_menu, main_opt, Normal, FALSE ).

skip_flag := FALSE;

IF (main_opt = board) THEN
  Market
ELSE IF (main_opt = next) THEN
  IF (main_opt = next) THEN
    BEGIN
      who := who + 1;
      Skip_Flag := TRUE;
      IF (Who > players) THEN
        IF (year+1 > MAX) THEN
          End_Game
        ELSE
          BEGIN
            Skip_Flag := FALSE;
            Update_Player;
            Update_Market;
          END;
    END
  ELSE IF (main_opt = pfolio) THEN
    IF See_Portfolio  ( It must get which player from Var. who )
    ELSE IF (main_opt = quit) THEN
      IF End_Game
      ELSE
        IF (main_opt = graphs) THEN
          BEGIN
            IF ( rst_flag = FALSE ) THEN
              See_Color
            ELSE
              See_Mono;
          END
        ELSE
          IF (main_opt = buy) THEN
            Buy_Stock
          ELSE
            IF (main_opt = sell) THEN
              Sell_Stock
            ELSE
              IF (main_opt = sb_info) THEN
                Get_Info;

IF ( skip_flag = FALSE ) THEN
  BEGIN
    GotoXY( 24, 22 );    { Center the "dialog" }
    InverseVideo;
    Write( '[RETURN] or <CLICK> to continue' );
    NormVideo;
    Cont_Menu;
    Clear_Screen;
    Curs_Home;
  END;   ( inner while - end of a game )

END;   ( outermost while - exit program )

Delete_Dialog( start_menu );
Delete_Dialog( stock_menu );
Delete_Dialog( main_menu );
Delete_Dialog( bond_menu );
Delete_Dialog( buy_menu );
Delete_Dialog( nam_menu );
Init_Mouse;
Show_Mouse;
Clear_Screen;

{ Say goodbye }
Replay( bye_pic, LSAVE, MODE15 );

  Exit_Gem ;
END ;
END.
```

(END)

Our Policy on Submitting Material

Contributions to **ST Applications** are welcome from everyone. We want a variety of articles and programs which can be helpful, useful or just fun for other Atari ST owners.

We prefer articles with accompanying programs (and hopefully, at least one graphic) which demonstrates a type of programming style or application. We wish to keep the content of the articles serious enough to enable readers to refer back to them in the future- something to build on. We feel this makes for a more meaningful relationship with the user and his/her computer. Chances are if a topic interests you, other people will find it useful as well. Our need for short articles is never ending.

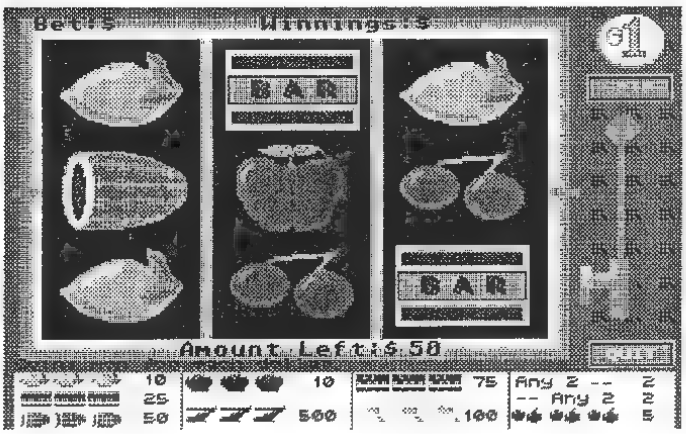
All articles submitted to **ST Applications** become the property of **ST Applications** unless otherwise agreed upon by both the author and **ST Applications**. Should you not receive acceptance from us within two months please call or write. Should you desire remuneration please so state with your submission. Our normal payment for articles begins with either a 1-year subscription or $35 (more, depending on topic, etc.). We pay $15 for software, hardware and book reviews.

Please include both text and program files on your submission disk. Be sure to send a hardcopy of both the article and program listing or graphic (including photos and drawings).

Our address:
ST Applications
P.O. Box 980
Forestville, CA 95436

Our Phone Number: **(707) 887-7879**

Super Slot Machine



by Roger L. Yap

Basic has traditionally been a language that has been looked down upon by most programmers in their quest for languages that can meet their needs in developing more "serious" applications. On just about every computer, the main criticisms have been its slowness and limited support of the special hardware features and qualities that make their machine unique. Many basics, while being easy to use, have been handicapped by their lack of flexibility and difficulty in debugging what is often spaghetti code. These complaints are equally applicable to the ST and until recently, languages such as 'C' or Pascal has been the only choice for writing programs which take advantage of the special abilities of the ST.

Enter GFA Basic which is a fast, full-bodied, but remarkably compact and efficient version of the language. GFA Basic not only has all the features of Basics for other computers (which often retail for several times the price), but it also offers a wealth of commands for accessing the graphics power of the ST. The bit-blit routines on the ST have traditionally been cumbersome to set up and use. A programmer must be very careful when using these routines as errors will usually cause the machine to lock up if one is not VERY careful in making calculations. Bit-blitting is important since it can be used to move portions of the screen from one part to another part or even off screen to a storage buffer in memory. Many programmers have chosen to avoid GEM entirely and write their own blit routines. Fortunately, GFA Basic programmers can avoid this task since the language offers an equivalent to the blit move routine. It offers another way of moving pieces of the screen, which is much easier to use and is just as effective. The Get and Put functions have been greatly enhanced over similar commands found in other machines as they can be used to operate on the screen. To define an area of the screen to be moved or copied, the Get command is used. It is used in the following manner and requires 5 input parameters:

```
Get xl, yl, xr, yr, var$
```

The values xl, yl, xr, and yr, define a square with the upper left corner specified by xl,yl and lower right corner defined by xr,yr. GFA "cuts" a duplicate copy of this screen area under the defined square and stores it in a string variable (in this case named var$) while leaving the original part intact. This screen block can now be moved back on to the screen at any location desired by using the Put command, which takes the following 4 parameters:

```
Put xl, yl, var$, writemode
```

where xl and yl are the upper left coordinates of the screen block, var$ is the name of the string variable used for storage and writemode used for special effects. It is normally set to a 3 to indicate an overwrite. These commands are simple to use; yet they are very powerful and were the main reasons that I bought GFA Basic. Using the Put command, it is now possible to "stamp" screen images anywhere on screen quickly and easily. One serious application of this function would be the redrawing of the screen after it has been drawn or written over. An application can draw lines, circles, text, etc. over a portion of its screen and can later restore what was there previously. This gives a programmer the freedom to create nearly limitless types of special effects such as the "undo" command found in most paint programs like Neochrome or Degas. The current screen that is being used for painting is duplicated and saved in a separate buffer that cannot be accessed by the user. If the user has made a mistake, such as wrongly placing a line or fill pattern, he presses the "undo" key and then the mistakes can be covered up by cutting pieces out of the spare screen and stamping over the mistakes. Thus, with few days of practice, even a programmer uninitiated to the ST will have NO problems writing educational, scientific, or recreational programs on the ST.

To demonstrate how these two simple commands can be used as a basis to write an entire application, I present Super Slot, which as you may have guessed from the name, is a slot machine program for the ST. All of the animation effects from the spinning of the slots to the pulling of the handle are all achieved by use of these commands. The Get and Put commands were used extensively to save portions of of the screen and redrawing them again at various locations.

Playing the Game

The Super Slot operates much in the same manner as an ordinary slot machine. You insert a coin and, if you are lucky, you will get more out than you put in. The amount that you get back depends upon the number of coins that you put it. For instance getting three water melons will result in 50 coins returned for every one put in. If you had placed 2 coins in, you will get 100 coins, etc. I felt like being a nice guy, so I allowed a spin to be taken without depositing any coins. Of course, if you do this, you will not be able to collect any coins. The buttons on the slot machine are clearly labeled and the left mouse button is used to activate all options. There is a $1 coin icon that is used to enter the bet. You can click several times on the icon to enter individual amounts or you can hold the mouse button to increase the entered amounts more rapidly. Then, simply click on either the "Spin" button or the slot machine arm. There is a "quick insert and spin" mode on this slot machine. As usual when you click on the $1 icon the amount bet will increase quickly. Now, without taking your hand off the mouse button, slide the mouse pointer over to the "Spin" button or to the top of the slot handle. Release the mouse button when you see the slot wheels spin. This should decrease the amount of wear and tear on your mouse. The amount bet, the amount won after each spin, and the total amount of money that you have left is always visible and, since this is the Atari ST casino that you are playing in, unlimited loans are available and it is possible to keep on playing even if you don't have any cash left. You will have a negative balance of course. When the game starts, you will be prompted to enter the amount of money that you want to begin with. I set the limit at $5,000 since allowing numbers in the the ten thousands or millions for this slot machine would be ridiculous. Please enter numbers only, as the program will not let you begin until you enter a number less than $5000. If you are lucky, you will hear some rousing sirens and you will be handed your coins on screen. I realized that watching 500 coins fall can be boring so I limited the number of falling coins to 20 in order keep the game fast moving. Hitting three items in a row gives you the added bonus of getting double sirens. One last thing: be sure to set your ST to low resolution.

Over the course of writing this program I discovered a small bug in v1.0 of GFA basic. Programmers who like to use Boolean logic will not get a correct result with the following:

```
w=5
ans = (w=5) * 10
print ans
```

since this will yield an answer of -10. To correct this, modify the routine to this:

```
w=5
ans = -1 * (w=5) * 10
print ans
```

which will now yield the correct answer of 10.

This slot program is fairly short, simple, and (I hope) straightforward. It features some really good sounds that took quite a while to program. This program has been tested on a Mega ST and its speed has been increased greatly. In fact, it was so fast on a friend's Mega 2 that I am thinking about renaming it "Blazing Slots".

MORE...

SUPER SLOT MACHINE

```
' /*  Super Slot Machine   */
' /*  by Roger L. Yap, Jr. */
' /*  Completed June 1987  */
' /*  Written in GFA Basic */
'
' * Check resolution and exit if not in low *
If Xbios(4)<>0
  Msg$="This program works|in low resolution only"
  Alert 3,Msg$,1," OK ",B
  End
Endif
'
' * Declare Array to hold various parts of the screen such as lemons, etc. *
Dim Shape$(8)
'
' * Save Old Color Palette so that it can be restored upon exit *
Gosub Getclrs
'
' * Set Palette to new values *
' * Read RGB data and set each color accordingly *
Restore Rgb
For I=0 To 15
  Read R,G,B
  Setcolor I,R,G,B
Next I
'
' * Load in title screen *
Bload "Dat2",Xbios(2)-128
'
' * Cut out the slot arm animation pieces from the title screen
' * This eliminates the need to paint a separate picture for
' * Arm movement.
Get 64,15,101,119,H1$
Get 135,15,172,119,H2$
Get 204,15,241,119,H3$
'
' * Opening sound effect *
For Ef=2 To 4
  For Eff=1 To 5*Ef
    Sound 1,12,Eff,5,1
    Sound 1,12,Eff+4,4,1
    Wave 3,3,0,8000,2
  Next Eff
Next Ef
Sound 1,0
Deftext 2,0,0,6
'
' * Get the starting amount of money *
Gosub Money
' * Load in slot machine picture *
Bload "dat1",Xbios(2)-128
'
' * "Cut' out slot pieces from opening scenes *
Gosub Images
'
' ******************************
' *** Start of main program ****
' ******************************
'
' * Do some house cleaning like changing mouse shape and setting text modes. *
Defmouse 3
' * Do an initial spin to start things going  And say hello to user. *
Gosub Spin
Alert 1,Hello$,1," OK ",B
'
' * Monitor user's mouse actions and act when the left button is pressed. *
Do
  Mouse Mx,My,Butn

    * Check if the user has clicked on the "Spin  button *
  If Mx>=274 And Mx<=316 And My>=31 And My<=75 And Butn=1
    Gosub Movehandle
    Gosub Spin
    Gosub Checkwin
  Endif
  '
  ' * Check if the user has clicked on the "Quit  button *
  If Mx>=275 And Mx<=317 And My>=157 And My<=171 And Butn=1
    Alert 1," Quit?  Are you sure?" 1 "Yes|No",B
    If B=1
      Gosub Putclrs
      End
    Endif
  Endif
  '
  ' * Check if the user has clicked on $1 icon and "inserted" a coin *
  If Mx>=279 And Mx<=315 And My>=1 And My<=29 And Butn=1
    Gosub Bet
  Endif
Loop
'
' ** Animation sequence for handle movement **
' * Stamp handle pieces on top of each other to create the
' * the illusion of movement
'
Procedure Movehandle
  Put 277,49,H1$,3
  For Ii=0 To 1000
  Next Ii
  Put 277,49,H2$,3
  For Ii=0 To 1000
  Next Ii
  Put 277,49,H3$,3
  For Ii=0 To 5000
  Next Ii
  Put 277,49,H1$,3
Return

' ** Fill the shape array with the 'cut" screen  shapes **
Procedure Images
  Bar=1
  Bbar=2
  Apple=3
  Lemon=4
  Seven=5
  Cherry=6
  Melon=7
  Bell=8
  Get 21,16,91,61,Shape$(Bar)
  Get 101,16,171,61,Shape$(Bbar)
  Get 181,16,251,61,Shape$(Apple)
  Get 21,63,91,108,Shape$(Lemon)
  Get 101,63,171,108,Shape$(Seven)
  Read Bx,By,Bx1,By1
  Get Bx,By,Bx1,By1,Bet$
  Restore Win
  Read Wx,Wy,Wx1,Wy1
  Get Wx,Wy,Wx1,Wy1,Winblt$
  Restore Left
  Read Lx,Ly,Lx1,Ly1
  Get Lx,Ly,Lx1,Ly1,Left$
  Get 117,120,150,147,Doll$
  Restore Botmid
  Read Bbx,Bby
  Put Bbx,Bby,Shape$(Cherry),3
Return
'
' * Check to see how much 'money' the user has "inserted" into the slot  *
Procedure Bet
  Put Bx,By,Bet$,3
  Put Lx,Ly,Left$,3
  Add Bet,1
  Dec Start
  ' * Write the bet amount info onto  the screen *
  Graphmode 2
  Text 57,10,Str$(Bet)
  Text 192,164,Str$(Start)
  Graphmode 1
  '
  ' * Clinking sound of a coin going down the coin slot *
  For J=6 To 8
    Sound 1,7,J-2,7,1
    Wave 3,3,0,8000,2
  Next J
Return
'
' ** Animation sequence for the "spinning" of the slot wheels. **
'
```

no way to build → X

Endifs not nested properly.

```
' * Randomly draw shapes on the screen using the pieces previously *
' * cut pieces.
Procedure Spin
  For Turn1=1 To 4
    Sound 1,12,Turn1,4,1
    Rate=1
    Restore Topleft
    Gosub Draw
    Restore Topmid
    Gosub Draw
    Restore Topright
    Gosub Draw
    Restore Midleft
    Winflag=1
    Gosub Draw
    Restore Midmid
    Gosub Draw
    Restore Midright
    Gosub Draw
    Restore Botleft
    Gosub Draw
    Restore Botmid
    Gosub Draw
    Restore Botleft
    Gosub Draw
  Next Turn1
  Gosub Music
    Gosub Draw
    Restore Topright
    Gosub Draw
    Winflag=1
    Restore Midmid
    Gosub Draw
    Restore Midright
    Gosub Draw
    Restore Botmid
    Gosub Draw
    Restore Botright
    Gosub Draw
  Next Turn2
  Gosub Music
  For Turn3=1 To 3
    Sound 2,12,Turn1,5,1
    Sound 1,12,Turn3,6,1
    Rate=3
    Restore Topright
    Gosub Draw
    Winflag=1
    Restore Midright
    Gosub Draw
    Restore Botright
    Gosub Draw
  Next Turn3
  Gosub Music
Return

' ** Stamp the cut pieces onto the screen **
Procedure Draw
  ' * Randomly generate index numbers for the shape array *
  T=Int(Rnd*8)+1
  If Turn1=4 And Winflag
    Winflag=0
    W1=T
  Endif
  If Turn2=3 And Winflag
    Winflag=0
    W2=T
  Endif
  If Turn3=3 And Winflag
    Winflag=0
    W3=T
  Endif
  Read X,Y
  Put X,Y,Shape$(T),3
  For Dd=1 To 20*Rate
  Next Dd
Return

' Did the user win something?  Find out!!! Make a LOT of noise if
' he did.

Procedure Checkwin
  ' * Check for two in a row at either end *
  Graphmode 2
  If W1=W2 Or W2=W3
    Coin=Bet*2
  Endif
  ' * Check for three in a row * '
  If W1=W2 And W1=W3
    Coin=-1*Bet*-(W1=Bar)*25+(W1=Bbar)*75+(W1=Apple)*10+(W1=Lemon)*10
  Put Wx,Wy,Winblt$,3
  Put Lx,Ly,Left$,3
  ' * Write the results on the screen *
  Text 206,10,Str$(Coin)
  Text 192 164,Str$(Start)
  Graphmode 1
  If Coin<>0
    Gosub Win
    If Threeflag=1
      Threeflag=0
      Gosub Win
    Endif
  Endif
  Bet=0
  Coin=0
  Put Bx,By,Bet$
Return
'
' ** Special sound effects **
Procedure Music
  For Mu=1 To 12
    Wave Mu,Mu,10,5000,1
  Next Mu
  Sound 1,0
  Wave 0,0,0,0,0
Return
'
' ** Flash the screen and make some noise in case of a win **
Procedure Win
  Xx=27
  Yy=68
  Xxx=244
  Yyy=104
  Get Xx,Yy,Xxx,Yyy,Old$
  ' * Draw a series of shrinking boxes *
  For N=1 To 6
    For Nn=11 To 15 Step 2
      Setcolor Nn,Int(Rnd*15)+1,Int(Rnd*15)+1,Int(Rnd*15)+1
      Color Nn
      Box Xx,Yy,Xxx,Yyy
      Dec Xxx
      Add Xx,1
      Dec Yyy
      Add Yy,1
    Next Nn
  Next N
  Deftext 13,4,0,32
  Graphmode 2
  Text 41,95,"Win!Win.Win "
  Get 27,68,244,104,New$
  ' * Cycle through the screen colors to simulate flashing and
  ' * create some siren-like noises.
  Nn=11
  For Ef=2 To 4
    For Eff=1 To 5*Ef
      Setcolor Nn,Int(Rnd*15)+1,Int(Rnd*15)+1,Int(Rnd*15)+1
      Sound 1,12,Eff,5,1
      Sound 1,12,Eff+4,4,1
      Wave 3 3,0,8000,2
    Next Eff
    Add Nn,2
  Next Ef
    Read R,G,B
    Setcolor I,R,G,B
  Next I
  If Coin>0
    Gosub Show
  Endif
  Graphmode 1
  Put 27,68,Old$,3
  Deftext 2,0,0,6
```

MORE...

```
Return
'
' * Draw winning coins on the screen *
Procedure Show
  Graphmode 2
  Lsx=29
  Limit=Coin
  If Coin>20
    Limit=21
  Endif
  For Ls=1 To Limit
    Put Lsx,73,Doll$
    For Js=1 To 3000
    Next Js
    Add Lsx,9
    For Js=9 To 10
      Sound 1,7,J,7,1
      Wave 3,3,0,8000,1
    Next Js
  Next Ls
  For Jj=1 To 3
    For J=6 To 10
      Sound 1,7,J,7,1
      Wave 3,3,0,8000,4
    Next J
  Next Jj
  Graphmode 1
Return
'
' ** Get the starting amount of money that the user has **
Procedure Money
  Start=0
  Print At(32,24);
  Input Start
  If Start>5000
    Gosub Mistake
  Endif
  On Error Gosub Mistake
Return
'
' * Correct input if user entered anything else but a number *
Procedure Mistake
  Start=0
  Print At(32,24);"      "
  Print At(32,24);
  Input Start
Return
'
' ** Save original pallets for later restoration,  people hate it when  **
' ** programs do not do this.
Procedure Getclrs
  Dim Sav%(15,3)
  For X=0 To 15
    Dpoke Intin+2,0
    Vdisys
    Sav%(X,0)=Dpeek(Intout+2)
    Sav%(X,1)=Dpeek(Intout+4)
    Sav%(X,2)=Dpeek(Intout+6)
  Next X
Return
'
'  ** Restore original colors and make everyone happy  **
Procedure Putclrs
  For X=0 To 15
    Dpoke Contrl,14
    Dpoke Contrl+2,0
    Dpoke Contrl+6,4
    Dpoke Intin,X
    Dpoke Intin+2,Sav%(X,0)
    Dpoke Intin+4,Sav%(X,1)
    Dpoke Intin+6,Sav%(X,2)
    Vdisys
  Next X
Return
'
' ** RGB color data values **
'
Rgb:
Data 0,0,0
Data 7,0,0
Data 4,4,5
Data 6,6,0
Data 7,7,0
Data 1,5,1
Data 0,7,0
Data 4,5,7
Data 7,7,3
Data 0,5,7
Data 0,2,7
Data 0,0,7
Data 5,4,0
Data 6,6,7
Data 5,1,0
Data 7,7,7
'
' ** Slot figure coordinates for placement of screen pieces **
Topleft:
Data 21,16,91,61
Topmid:
Data 101,16,171,61
Topright:
Data 181,16,251,61
Midleft:
Data 21,63,91,108
Midmid:
Data 101,63,171,108
Midright:
Data 181,63,251,108
Botleft:
Data 21,109,91,154
Botmid:
Data 101,109,171,154
Botright:
Data 181,109,251,154
Set:
Data 189,157,255,166
```

END

APL

Is This What You Have Been Waiting For?

by Bruce E. Wiebe

What I like best about my ST is the variety of good quality, low cost programing languages for it. *The Spencer Organization* has added one more to the list by dropping the price of its APL package from $295 to only $95.

Does this mean you should rush out and buy it? Maybe, maybe not. Read on and find out. What I plan to do is tell you what APL is all about and what its strengths and weaknesses are. Spencer has made the choice easier still by providing a demo version as public domain. It has all the features of their regular package but you can't save your work. Of course you don't get any documentation with it either. See your users group and get yourself a copy.

Trying to explain the concept behind APL programming is not an easy task. It is unlike any of the "conventional" languages like Pascal, C, COBOL, and BASIC. Looking at APL code is not unlike looking at a bunch of Greek symbols. There is a very good reason for this, APL started out not as a programming language (ironically that is what APL stands for) but a mathematical notation.

Anyone with a math background knows that mathematical notation is very terse, very precise, and until you get used to it, very impossible to understand. Here is where the Greek symbols come in. APL uses several dozen Greek and other symbols to perform its operations. To confuse things even more, these symbols perform different tasks depending on whether there are values on both sides of the symbol (dyadic) or if there is only a value on the right-hand side of the symbol (monadic). Since APL performs its order of execution in a right to left manner with no consideration for order of operations, it is very easy, and tempting, to cram a lot of information on one line. This makes for quick coding but a nightmare for maintenance and debugging.

I will be the first to admit that this free and easy programming style can quickly get you into trouble but APL does have its uses.

At my office (I work in the insurance industry), it is now policy that all actuarial students have a knowledge of APL. This is because APL is so powerful at manipulating statistical data. You can multiply matrices, add the rows together and do almost anything you want when it comes to numbers. All in only a few keystrokes. And since APL is based on mathematical notation, the actuarial types feel quite comfortable with it.

APL is interactive. As soon as you type something, you get a response. No compile-link-edit cycle here. Declaring variables is a thing of the past as well. Vectors can be changed to matrices, and in turn changed to four, five, or any dimensional array with a keystroke.

Continued on page 29 ==>

LateST News

by Stephen Mortimer

The reorganization of Atari, as described briefly last month, is much more extensive than originally thought. Atari Corp. is now the parent company of Federated, Atari Computer, and the Entertainment/Electronics division. Chuck Babbit, former Vice-President of Ashton Tate, is now the President of Atari Computer. Tony Gould, who worked in sales at IBM for ten years, is VP of Sales for Atari Computer. Atari Computer has hired Winston and Winston, a public relations firm, to cover their computer business, and is looking for an advertising agency.

Atari Computer has cut distributors out of the picture, and now sells directly to their dealers through Atari employed sales representatives. Several mail order companies are not being allowed to sell any Atari hardware as a result. The dealer chain is being strengthened greatly and Federated stores now carry the Mega ST. Atari's main thrust is aimed at the business market.

Additionally, Atari has created a telemarketing group, shuffled around management, and is in the process of hiring more marketing people. There is now a dedicated newsletter editor at Atari who is creating dealer, developer, and user group newsletters on a regular basis. Television advertising for the ST will begin in the third quarter of 1988, according to Atari. On to the news!!

Enhanced Resolution for the Mega

The Mega ST will have two enhanced resolution boards available for it by the end of the summer. One from **ISD**, and the other from **Moniterm**. The board from ISD will connect to the Mega's internal bus and allow a maximum resolution of 1280*960 in monochrome, and 640*480(70Hz) and 832*560(50Hz) in color. Both color resolutions will allow up to 16 colors to be displayed out of a palette of 262,000 colors. Shown at COMDEX on a 19" monochrome monitor, the display was crystal clear at the 1280 by 960 resolution while running Calamus, an upcoming DTP program. A delivery this summer is expected at an undisclosed price.

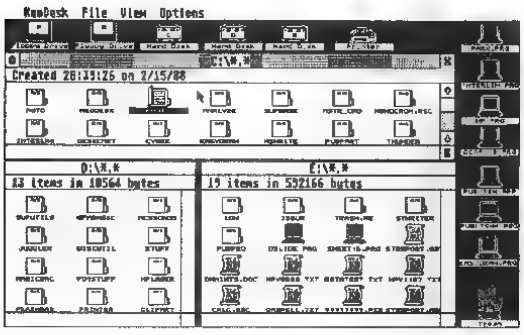
The other company working on a monitor board is Moniterm, a giant in the monitor industry. After seeing a Mega at COMDEX running the ISD board, the President of Moniterm reportedly decided to have his company design one. Atari is working closely with Moniterm to ensure it is finished by the 6 weeks deadline set by Moniterm at COMDEX. The Moniterm board will allow a 1280*960 monochrome resolution. If Moniterm follows previous price patterns, they will sell the board with a 19" monitor for about $2000. If you are serious about desktop publishing, these will make your life a lot easier!

NeoDesk

NeoDesk is a complete desktop alternative for the ST from Gribnif Software. The desktop is more graphically based than the current ST desktop, yet allows much more to be accomplished. First of all, the desktop allows up to seven windows to be open at once. Each may have its own search template (eg. *.ACC) that determines what files the window will display. Unlike the normal desktop, files are shown only as icons in the directory window. To alleviate the problem of scrolling both horizontally andvertically through icons, NeoDesk formats the icons in the window so that horizontal scrolling is not needed. Files may be

organized by date, size, type, or according to their position on the disk. Sort by type arranges program icons first and data files second. Program icons are little computers which can be easily distinguished from data files. Disk icons are shaped like floppy disks, hard disks, or RAM chips for a RAM disk. When a file in the window is highlighted, its size, date, and read/write status is displayed in the window status line. If multiple files are highlighted, their total size is shown..

Other than looking better than the normal ST desktop, NeoDesk has more features. Up to ten file icons may be situated on the desktop and executed from it at any time. The actual programs aren't on the desktop, but each icon is associated with the program's path name which is used for execution. Disk copying can be accomplished by copying all sectors and formatting the disk, just copying all sectors, or copying all the files on the disk. Batch files can be executed and the shell of your choice run automatically from NeoDesk. Keyboard equivalents are available for almost every command in NeoDesk, from Show Info to renaming a file. Show Info for a disk provides a very detailed report including sectors, tracks, hidden files, and an editable volume name. This same function used on a file allows that file to have its creation data set to the current date and time so files can be rearranged. Disk formatting options include over 80 tracks and both 9 and 10 sectors. Other features include application installation, environment variables, hard and soft boot key combinations, and showing files to the screen.



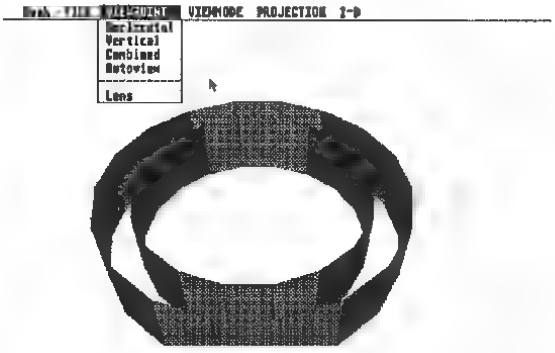
Neo Desk

Included with NeoDesk is an icon editor that allows all program, data, folder, disk, printer, and trash icons to be user defined. Even the desktop pattern may be changed. NeoDesk provides a printer queue accessory that will output files in the background when they are dragged to the printer icon. Files in the queue may be rearranged or removed from the queue. NeoDesk requires 145K of memory and is $29.95 from Gribnif Software, PO Box 350, Hadley, MA 01035.

Two New CAD Programs

Master CAD and GFA Draft Plus are two recent releases from MichTron that bring the ST two more quality design packages. Master CAD is a 3-D modeling program and GFA Draft Plus is a 2-D drafting package.

GFA Draft Plus is an upgrade to the popular GFA Draft released over a year ago. Draft Plus offers a work area of 51 inches square that can be scrolled through with the GEM scroll bars. Macro drawing commands allow pictures to be drawn or plotted through the use of a programming language similar to LOGO. A whole drawing may be created with the language, drawn, and output to a plotter or printer in GFA Draft Plus. GFA BASIC can be used with Draft Plus to compute vectors for a drawing. 3-D drawings can be loaded into Draft Plus from GFA Object if the file is saved in the macro programming language format. Drawing features offered by Draft Plus include the ability to draw circle andellipse segments, rounded corners, 3 point arcs, lines at specific angles to each other, circles, and rectangles. GFA Draft Plus also offers parallel and tangent line drawing modes, polar coordinate, absolute, and relative measuring systems, three different font styles, and the ability to create regular polygons by creating a circle and defining the number of corner points. A comprehensive manual is included with the program as well as a disk full of object libraries. $149.95.



Master CAD

Master CAD is a different breed of a CAD package. Rather than being a professional wire-frame modeling package, it allows a two dimensional design to be projected into a three dimensional object that exists between or on two planes that can be user defined. Once the object is in the 3-D, it can be observed from any various viewpoints and viewangles, both external and internal. The object can be moved, copied, flipped horizontally and vertically. Its proportions can be changed and surfaces seen as solid or transparent. Complex spun objects can be created around a user-defined-axis. Auto dimensioning calculates the distance between any two points in a picture. Your masterpiece can be output to any GDOS supported printer or an HP Color Pro, HP 7550, or

HP 7570 plotter. I have found Master CAD easy to use, and is similar to CAD 3-D from Antic. You can create a unique 3-D object with Master CAD in a short amount of time using the well implemented GEM interface. The list price is $249.95 but it is on sale for $199.95 from MichTron at this time. MichTron, 576 South Telegraph, Pontiac, Michigan 48053. (313)334-5700.

COMDEX Report

Many of the products mentioned in this column were shown or introduced at the spring COMDEX in Atlanta held during May.

Xetec was displaying a cartridge-based floating point math co-processor called **XCEL**. It will sell for $199 and boasts speed improvements in mathematical calculations of up to 30 times. Custom math libraries must be incorporated into any program using its capabilities.

Other products shown included **GFA BASIC 3.0** that is scheduled to be released in July and includes 300 new commands and greater speed. SBT was showing a version six of their Database Accounting Library that will be on sale shortly. Several upcoming desktop publishing packages were shown including Publishing Partner Professional, Calamus, and Home Publisher from Robtek. Atari's main theme was desktop publishing, MIDI, and business software. No new hardware was shown except for the CD-ROM. Atari Computer has recognized that the ST is going to be their bread and butter for a long time to come.

Product Previews

The **Revolver** is a new product from Intersect Software, the makers of Interlink ST and Music Construction Set. It will allow a snapshot of the computer's memory to be taken and saved to disk. This file will contain all the desk accessories and auto folderprograms loaded at the time it was saved. Even RAM disk contents will be intact when the file is "rolled" into the computer at a later time. Games can be saved and continued at a later time. If you need to run another program while in another, "roll out" the current program, "roll in" your new program, and then go back to your original program exactly where you left it. The Revolver has timed saves, full disk commands, a VT-52 emulator, control-panel options, a snapshot program, a reset-proof RAM disk, a printer spooler, and a 40 folder fix. Most of these features are available when you call Revolver from within your program. It will be available in late June at the price of $49.95 from Intersect Software. (800) 826-0130.

Migraph will be releasing a **flatbed scanner** early this summer at a price less than $1000. It will connect to the parallel port and require one megabyte of memory to operate. The scanner can scan a full page at 200dpi (it's maximum resolution) in ten seconds. It can also act as a photocopier with its built-in thermal printer. Later this year, Migraph will also release a resolution independent IMG paint program. Migraph, 720 South 333rd Street, Suite 201, Federal Way, Washington 98003. (206)838 4677.

Tidbits and Rumors

A substitute for GDOS is about to be released, but it's not from Atari. Dubbed **GDOS_LP** for GDOS Liberation Front, the new GDOS optimizes the original code. This will alleviate any slowdown that some users find when using GDOS. Also, it will allow a new ASSIGN.SYS to be loaded in and recognized without the computer being re-booted. This can be accomplished through the use of a companion desk accessory. It must be done outside of a GDOS based program. Future versions will have a built in font editor and selective loading of fonts, according to Charles F. Johnson, one of the program's authors. It may be released as shareware in the near future.

A new version of **TOS** has recently begun beta-testing world wide. The new version includes lots of improvements, many of which the average user will notice. Some of the features added:

- A more IBM compatible disk format
- A move file feature
- The ability to autoload GEM programs
- A new file selector with 16 drive buttons
- The ability to rename folders
- Improved disk performance
- File copying using all available memory
- Interruptable Copy/Delete/Move functions
- The 40 folder bug is fixed
- Soft and hard boot keyboard combinations

Deskset, Atari's upcoming desktop publishing package will sell for a price between $500 and $1000. Release one is orientated towards people familiar to the Compugraphic typesetting commands and will be released this summer. Release two is being developednow and will use the GEM environment extensively.

Federated stores in the Southwest are currently being transformed into full Atari dealerships. Several are already carrying the Mega ST and Atari Laser. They also have a wide variety of professional level software and will soon begin to service Atari computers.

Ultrascript, the Postscript clone being developed by Atari and Imagen will come packaged with the Lucida type family when released. It will require a minimum of two megabytes to operate and will operate at a speed comparable to the Apple Laserwriter.

The **"ST Accelerator"** board for the 520 and 1040 ST should be shipping by now at a retail price of $250. A board for the Mega is currently under development. Strange Systems are also creating an eight card expansion box for the ST that will be called the **"ST Expander"**. Possible uses for it are networking, memory expansion, PC emulation, and enhanced resolution boards. It is scheduled for release at the fall COMDEX in Las Vegas.

At the recent spring COMDEX in Atlanta, Sig Hartmann of Atari stated that **125,000** STs had been sold in the US and over **500,000** abroad.

Atari Computer is sponsoring a summer sale for the MEGA and Atari Laser printer. If a dealer purchases one MEGA and one Laser, they will get a price break and receive Microsoft Write, Timeworks Desktop Publisher, and the VT100 cartridge for free. Dealers say that the price of a total system will be reduced greatly due to the sale. If you want a MEGA and Laser, now is the time to buy it!

Corrections

The address posted with the Universal Item Selector was wrong. It should have been A&D Software, c/o MacDonald Associates, 226 NW "F" Street, Grants pass, Oregon 97526. A new version of UIS will be available at the beginning of June for four dollars and the return of your master disk.

Symmetry, the CADD package in the works at ISD has changed its name and price. It will now be called DynaCADD and will sell for $550. ISD has signed up several Value Added Resellers that are actively promoting DynaCADD before its release. It has receives very good reviews from anyone who has seen it and may be released in June.

Next month, Superbase Professional will be the main feature. Here's a quick peek at it for you. It has a relational programming language, forms editor, enhanced report generating ability, and an extensive query capability.

END

More Letters from page 5

```
#include <stdio.h>
main()
{
 int i;
 char spane = 130;
 char spano = 162;
 char symbol = 189;
 printf("C%crdoba, Ver.,
M%cxico\n",spano,spane);
 printf("%c Copyright\n",symbol);
 }
```

My problem has been in terms of VDI and the concatenation of characters and strings. What can I do to print in the screen a string with different ASCII characters while using the v_gtext function?

Another problem I have met in C is sending whichever kind of characters to the printer. Which makes me ask, "What happened to your tutorial section in C?"

Once again, thank you for helping find any solution to my problem.

Vicente O. Zanatta Tress.
Cordoba Ver., Mexico

END

More APL from page 25

Contrary to popular belief, **APL** is not difficult to learn. You can do something productive in the first 5 minutes. Why? Because you don't have to know much to get started. If you don't need a function, you don't have to learn it. Try writing a COBOL program without understanding the DATA DIVISION.

As mentioned, **APL** is interactive but, it is also interpreted. This means that execution is *s-l-o-w-w-w.* I would **NOT** recommend writing any CPU intensive applications in it. ST BASIC is also interpreted. Compare ST BASIC with LDW for a comparison of relative speeds for compiled and interpreted languages.

Speed is by no means the biggest drawback of the language. Depending on your application, speed might not be important at all. If you spend most of the time staring at the numbers on the screen or typing in data, it is obvious that a few milliseconds here and there are not important.

The biggest problem is maintenance. If you ever try to change someone else's **APL** code, you might as well start from scratch. **APL** is a write-only language. **APL** die hards will tell you that is not the case but I have yet to see someone else's **APL** code that I can understand, and I've been using if for over three years. (I often can't understand my own **APL** code after a year or two).

My recommendation? Stay away from **APL** for commercial applications. If you are a writing a quick routine for your own use, **APL** just might be what you are looking for. If you have a slew of numbers that need analyzing ASAP, there is absolutely no contest, get APL; pronto!

END

Freeware/Shareware

by Donovan Vicha

If you haven't already booted this month's supplementary disk, you might want to give it a try. Go ahead, I can wait ...Well, maybe you're not into the Three Stooges. Quite frankly, neither am I but then I've designed some books about the Stooges and may be prejudiced. But wasn't it perfect? -- No mistake, that was Curley. I'm beginning to like the ST's sound chip more and more now that digitized sounds are being used (Dungeon Master, Star Trek, etc.) on the ST. The AUTO folder contains a program called Autoplay, an .SND file, and BASCODE.EXE that carries the GFA Basic code for the program. Due to space restrictions on the disk, I did not include a Degas or Neo picture file that could also be displayed while the sound is performed, but the program is set up to do so.

Also on the disk should be a folder with the rest of the goodies shareware author David R. Sullivan has created: Playit.prg, for loading and playing SND files; playedit.prg, for converting ST Replay .SLP files to .SND and for changing speeds; and another .SND file that is excellent for trying out the edit program. I would have liked to include others from my growing collection, which include the "Battlestations!" announcement and "Space ..." from Star Trek (TV), terrific clips of HAL from the movie 2001, the Lone Ranger's immortal cry, the Batman theme, Darth Vader quotes, clips from Hitchhiker's Guide to the Galaxy, and Bugs Bunny. More can be had downloading from GEnie, but I warn you, these files can get pretty big (hence the relatively small files on disk). In fact, there is another sound player program that plays Digisound files, which share the same .SND tag (and are not compatible, causing awful sounds and sometimes a crash). These are enormous files, but are also excellent: the opening lines from The Outer Limits, quotes from Daffy Duck, and a demo named "Elrod1" that has to be heard to be believed. I plan to send Digiplay's author a DS disk so I can have the entire Outer Limits opening, which runs over 600K. But that's another area I'll leave for some other time.

Sullivan's Playit set of programs is a much more versatile and diverting piece of work. But be warned that some .SND files may not work with Playit. The Playit program opens to a screen that presents the author's name and user group affiliations, BBS, etc. and an Item Selector for loading a file to play. From a hard drive or ramdisk this is very fast, with a bit longer wait for disk accessing (depending on the size of the file, of course). Then you are given the choice of playing it again, loading another file, or quitting. The Autoplayer can be fully scripted to create picture and sound buffers, repeat the script file, set speeds and delays, or any number of GFAtype commands. Shortly, there will be an updated version, Autoplay II, which corrects a few bugs and adds faster disk I/O operations, reverse playing of

sounds, faster picture swapping, memorylimited buffering (limit of 5 pix right now). A Megamax C version is also in the works.

David Sullivan found it necessary to correct a major bug in GFA basic to clean up Autoplay. One of his commercial releases, **GFA Basic Developer's Toolkit #1** is an outgrowth of his bug hunting in the GFA Basic code. On it, he includes source code for his two shareware games, Space Intruders and Missile Defense, as well as useful routines for creating GFA Basic games. David has a lot of projects underway as well as under his belt. His company name is *Total Control Systems*, and perhaps his most notable commercial release is **The Celebrity Cookbook**, distributed by *Merrill Ward & Associates*, who are bringing a GEM environment to the 8-bit Ataris (and may also be adapted in some form for the ST). The Playedit program is a nifty kick. If you own ST Replay, and David thinks very highly of it over the other sound digitizers out there, you can use this program to convert its files for playing sounds with Playit and Autoplay. You can also change the speeds on the sound and then save the file, although unfortunately, this overwrites the old file. (You can also reload and get the sound back to its original speed, so it's not that unfortunate, I suppose.) Usually, the speed of the .SND files I've used were either 10 or 15 kHz, which are two of the 6 settings (the others: 5, 7.5, 20, and 31). The Goontime file can be manipulated to good effect, which is why I chose it to accompany the rest, going from the sultry woman's voice to one of the Chipmunks' voices at 31 kHz to foghorn from hell at 5 kHz.

David's other shareware contributions include those games mentioned earlier, plus some Colorburst utilities: Miniflip, 3D Merge, Grey Scale, and Flipmake, and Color Desk, for changing the colors of your desktop. Most of his ideas have come from members of his users group, SDACE. If you'd like to know what else he has on the burners, send a donation if you use any of his programs. He responds to every donator, often sending demo versions of his programs or updates, if requested. I received a three page letter as well as three more pages describing programs he has written for the XE as well as the ST. He has been an Atari addict for 8 years, starting with a 400 and working his way right through to an ST. That experience has given him a good savvy at finding and filling niches in the needs of users, and he really wants to encourage users to send him suggestions. Send your check ($10 basic donation, $7 more for code for Basic games, $15 for Megamax C games code, or $10 for nongame programs code) to David R. Sullivan, Total Control Systems, 4156 Tolowa Street, San Diego, CA 92117.

Oh, it might all seem like a clever waste of time, I suppose; something that just adds to the booting time. Even when the dreaded deadline doom is upon me, however, I kinda like having HAL 6000 telling that he is functioning perfectly. It makes me feel warm and mushy about my computer, my helpmate under fire.

A Really Deluxe Slideshow

John Brochu has some very good shareware programs that are useful and entertaining to a great many ST users, I'm certain. I was looking forward to having the excuse to download his MSPlay program and other people's music files, but discovered that I need a MIDI device for playback. Thus, I'm just going to focus this time on the one with which I'm most familiar, his DSlide (short for Deluxe Slideshow) program, now into version 2.0. I like this program not only for its ability to show a growing variety of art file formats, but for its excellent script file capabilities.

And now for a rather personal, confessional-type, digression. I used v1.0 of this software when I thought I was using v1.1, which is when scripting was introduced to the program. I intended to upload a slideshow tutorial I created using IMG Scan, DEGAS Elite, Tinystuf, and DSlide, but I was unable to get the .LST file to work (wrong version). This file is necessary to show the slides in proper order and with what was hopefully an adequate amount of time to read the captions on each slide before proceeding to the next one. I ended up using Tinyview even though it has no scripting capability to show them in order. I have a tough time getting my slides into the proper order so Tinyview can play them that way (I'm often stymied by what other people know is simple). So, I think the tutorial was uploaded defective in this manner and only later did I discover my mistake and upload DSlide (v1.1) and the necessary scripts. I find DSlide works very well in conjunction with Newsdisk, which I mentioned last month.

The latest version now supports Spectrum files (.SPC and .SPU), and features enhanced keyboard and mouse controls, which include a built-in Help screen, a quick-step mode with the abilities to reverse the sequence, restart, or designate first and last slides. As with v1.1, this program includes titling of slides which is a good feature, the switching from modes or resolutions is smooth and quick, and color animation support is available for DEGAS Elite pictures. And the script files, of course, have a few new additions as well, now allowing these parameters to be set on an easily composed .LST file: titling on/off, continue on/off, cycle or step mode, timing, pathname directions, and filename listings, which include the use of wildcards. The versatility of the scripting files is what makes this program stand out. For more than just creating a window display or a "home movie," one could create flash card shows and quiz games for the kids or business presentations and tutorials. With a donation to *Advanced Software*, you can feel free to experiment with this wonderful program and your slides or paint programs, which were bought also, in one way or another. I will be making great use of this program for a diskbased magazine I am working on. To spare the downloader the despair of seeing a

2200 block D/L go blotto after 1782 blocks, I will put separate slideshow components into separate files, with the Newsdisk, DSlide and scripts, and text files alone in another file. I could also assign the separate slideshow components to folders, making for neatness on the main directory. The author of this excellent program has made it known that there will be no further releases of PicSwitch, which is another case like Tinyview, everyone uses it but only 50 or so donations have been made. John is not losing his house due to such lackluster support (my words, not his) for he is fully employed as Production Supervisor/Senior Programmer for Digital Vision. He is willing to support his programs with enhancements and is online at GEnie (JAKOB), CIS [70376, 1235], and Delphi (JAKOB). If you use his programs, donate and send comments for improvements or suggestions for other programs. His address is: Advanced Software, 21 Northend Street, Peabody, MA 01960.

Yet Another Set Of Desktop Accessories

Shareware products for the IBM and Macintosh vary in quality in the extreme, but associations of shareware producers have become viable and have improved the reputation of the concept to such an extent that its a minor industry in and of itself, with some shareware products priced the same as high-end commercial products. As far as Atari computers go, there are very few high-end commercial products out partly because of the small U.S. market, but also because, let's face it, we're a low budget audience, too. Thus, it's not amazing that the good shareware products, programs that fill niches no commercial producer is going to the trouble to fill, are not reaping in thousands of dollars for their authors. Then again, there are niches that are overflowing ... If you took a full directory listing of the almost 7,000 files on GEnie, for instance, and threw a dart at it, chances are you'd either hit an adultrated file or a set of desk accessories file. **Deskpac Plus** is a fairly recent file that is available, and I'm afraid I cannot recommend it very highly. While it is visually the nicest such accessory I've seen it is slow in execution and offers nothing truly extraordinary in the way of accessories or in implementation. That I mention it at all is due to its creators effort to go the shareware route rather than attempt to market it commercially. The following text accompanies their program: Given the success of Red Ryder in the Macintosh shareware market, we here at *Advanced Environments* have decided to see if **DeskPac Plus** can be successfully distributed the same way. What this means to the end-user is a lower cost, since we have eliminated the high cost of paying a distributor, as well as saving the cost of reproducing paper documentation and pretty boxes. We made this decision after much debate, however, given the news from other shareware authors that their programs were being used but without receiving the usually small donation.

Since we are only asking $10 for this, our first product, we are hoping that users will recognize the advantages of distribution though the shareware market. Not only is the FULL WORKING copy of **DeskPac Plus** being offered, but with extensive documentation and COMPLETE SOURCE CODE! If you decide to download **DeskPac**, please send a check for $10 to: Steve Nies, Advanced Environments, 220 Treu Terrace, NW Palm Bay, Florida 32907 Good intentions. For the record, the accessories list of this package is as follows: digital-style clock, calender window, phonebook, extremely versatile calculator, appt. book, free ram window, note pad, copy and delete file, and return to desktop. As I said, it's the best looking accessory of its type. Until **DeskPac Plus** is brought up to par, with other, similar PD offerings, especially in the speed department, there isn't much here but a promising start (and a 95K .ACC). I am unable to comment on whether Turbo ST or a blitter improve the speed. I would think it's quite likely, so if you're so equipped, take a look at it. If you use it despite its slowness, then you still owe the authors a just donation. I do admire the faith of the author and encourage others to support the shareware concept either with your genius and code, or with donations and suggestions.

Coming Soon

Many of the accessory functions offered by **DeskPac Plus** are included in two new GEM type operating systems in a much more comprehensive way. I'll be covering them both, **NeoDesk**, which is commercial but promoted strictly through the nets (you might have to look for it under the Review Board) and **VDOS**, next month, when I've had more time to play with them, but I can say unequivocally that either of them will be replacing the GEM desktops of STs all over the place. And it's about time someone took care of the shortcomings of GEM as it is now implemented on the ST. The shareware network is where such things are being addressed and will continue with our support. Although there is allegedly a program that comes with MicroSoft Write that makes for easy writing of ASSIGN files, there is still a need for some kind of facilitator of this process for those inclined toward desktop publishing on the ST. I see something that could read a disk directory into ASCII, at least providing a pretyped file of the font filenames needed to write an ASSIGN file. A nifty accessory that works with GEM window oriented word processors, or a set of macros that could do it for WordPerfect is something I am investigating. Could this be a free shareware idea? Instead of a donation, send me your work! For that matter, send your ideas to this column via ST Applications. Or send comments on PD programs you'd like to see improved. I love E-mail on GEnie [D.VICHA].

Continued on page 39 ==>

The Independent Publication for Atari Business Computer Owners July 1988
Volume 1/No. 7

ST BUSINESS™

COMDEX Spring '88

THE NEW, NEW ATARI

One of the most profound experiences we've (Carole and I) ever had was to attend COMDEX Atlanta, May 9th/12th and observe the NEW, NEW ATARI in operation. Gone, are the "I'm bigger than you are, I'm better than you are" statements, the "soon to be released" products and promises that would certainly be extended again and again due to technical problems. A new professionalism has settled over the scene.

You'll remember, back when Jack and the boys took over from Warner Communications and decided to call it the NEW Atari, to differentiate from the old Warner operations. That was certainly the right statement at that time, no question.

They needed to be a little boastful and visible. After all, ATARI, as we now know it, didn't really exist.

Now, however, there's a new "deeds not words" approach. A low profile statement that for want of a better choice, we'll have to call the NEW, NEW ATARI. At Atlanta this Spring, there were no laboratory curiosities on display, no "soon to be delivered" products and no promises.

WYSIWYG (what you see is what you get - an acronym commonly used in DeskTop Publishing) was operating at full power in the ATARI camp. No transputer, no 68030 machine, no

(Cont'd on Pg. 8, Col. 3)

Low profile is the only way Atari's presence at the annual Spring rite of COMDEX, Atlanta can be described. The booth was ample, the software all business, and there were no games. Quite a contrast from previous shows.

Using the standard ATARI layout and equipment, the booth consisted of a number of stand alone pedestals on which Megas and 1040STs were running a variety of software. This proven display allows dealers to circulate, view the various software products and receive a demo from a knowledgeable individual, in many cases the developer himself.

ACTIVITY WAS GOOD

Top management stuck around for the whole show and appeared very pleased with the response and activity at the booth. In addition, of course, Neil Harris, Mel Stevens, Arnie Waldstein, and a number of people from ATARI customer service were on hand to keep things running smoothly.

President Sam Tramiel flew in for the Monday opening and apparently went back to Sunnyvale on the next flight. That's having confidence in your management.

A highlight was the Atari Computer programmers party at the Mariott Marquis on Tuesday evening. Management vowed to continue support of the programmers' efforts and build Atari Computer to a bigger and better organization. Mention was made of new dealers and

(Cont'd on Page 3, Col. 1)

ST BUSINESS

EDITORS & PUBLISHERS
Don & Carole Terp

CONTRIBUTING EDITORS
Bob Jones
Eric Neilsen

ST BUSINESS is published as a service to ATARI ST computer owners.
Send all items for publication to:
ST BUSINESS
5140 Appletree Drive
Roanoke, VA 24019
or phone: 703-977-1058

INDEX

EDITORIAL

We'd all like to believe major companies run smooth, are well planned and have every contingency accounted for. It's not true of General Motors, General Electric or General Trash and Hauling.

Tugging and pushing by outside forces can ruin a manager's day. In fact, if there were no outside forces, most managers wouldn't be worth near what they're paid.

Let's take two in the computer industry for example; The 256K and 1 Meg DRAM chip shortage, caused by a chip dumping incident where the Japanese were caught red handed, and the very low price of the US dollar on the foreign currency markets. Both of these anomalies are creating serious problems in world markets.

The chip embargo would have had positive results had Congress acted when originally requested by industry. But, US manufacturers got out of the 256K and 1 Meg chip business when it became unprofitable due to the very low Japanese prices.

After there was no US production, Congress put a stop to Japanese dumping by embargoing its chips.

Hence the chip shortage, and a cloud over a very hot industry that will get hotter as the economy turns down. Most "experts" are saying September for an end to the dry chip well.

Another factor that has driven our economy along has been our exports and foreigners vacationing in the US cheaper than they can in their own countries, due to the devalued dollar.

Japanese vacationers invaded Hawaii years ago, our West Coast within the last five years and are now moving East to New York and Washington, DC. Meanwhile Europeans are vacationing at bargain prices with their long lost cousins in NY, Florida and points West.

With new "liberal" visa policies Warsaw or Moscow may make an offer for Chicago or Minneapolis. No chance for Florida, Russians and Poles like colder climates.

All this while US manufacturers are having a field day shipping everything "under the sun" overseas. For once we can underprice everyone, everyone except the Japanese.

Japanese imports are restricted by a convoluted system of rules and regulations that make competition from the outside next to impossible. In fact the Japanese have been known to create import specifications to match its manufacturing processes even though a foreign supplier has a superior product. That's government/industry cooperation unheard of in the US.

If we break into the Japanese market with our technical innovations now, we'll damage its finely tuned economy. We'll wreck its protective tariff and government subsidy system and start money flowing in the opposite direction.

Since Japan has few natural resources, it's very dependent on adding value to everything it imports (this is its major source of income). An imported finished product can have no value added. Therefore, the Japanese can't import any quantity of finished goods, it can only export.

This will eventually produce a very severe economic confrontation, not only with the US, but with all major money centers, particularly if the world economy goes into a recession or depression as many have predicted. The Japanese know this and are very guarded. They have much to lose by any trade balance that would place them in the role of an equal trading partner with the US.

US Navy Commodore, Matthew C. Perry opened up Japan by negotiating a trade treaty that was signed, in Tokyo, 135 years ago, on July 14th, 1853. Perry's great white fleet (all US warships were painted white, a symbol of peace) started it, but little has happened since.

However, the World is changing, and we must change with it. Remember the dinosaur, he didn't. ▲

COMDEX SPRING '88. . .

(Cont'd from Pg. 1, Col. 3)

promises to sign up over 600 by the end of the year.

The food was excellent and the usual gang was friendly and full of fun. In addition to the software developers who manned the booths, there were a number of people from South Eastern Atari UGs who were manning Atari's portion of the booth, and relieving the various developers as well as aiding in set up and take down tasks.

Shawn Fogel, President of SoftLogic, demo'd Publishing Partner Professional for one and all. There are so many new features in PPP that, when the release comes in about 5 weeks, our review will have to be in two parts.

According to Sha n, SoftLogic has tried to stay one step ahead of the competition by being innovative and progressive in its development of PPP. Actually it's as good if not better than Aldus' Pagemaker at $795, and Pagemaker doesn't run on the ST!

Professional will list for $199.95. An upgrade from PP version 1.03 will be $50.00 until Professional ships and then $99.00. Get your money in early and save $40.00!

VIDEO ON THE ST

Antic Software demo'd its Cyber series of programs at two different stations. They attracted a crowd with 3D animation (3D drawing, no glasses required). We also learned that John Russel Innovations (a company), will have a GenLock PC board for the MegaST in July and 1040/520 versions during October.

GenLock permits the superimposition of computer graphics over or in place of conventional video images recorded on your VCR. You can now give Walt Disney or Hanna and Barbera a run for their money with home animation or simply put commercial titles on your home video tape. What a market!

We, here at ST BUSINESS are into video in a big way and can vouch for the pent-up demand for good products in this area. A little promotion in the right places will certainly capture this lush market similar to the way Atari has gained recognition in the music industry with built in MIDI ports on the ST.

This is also an area that Amiga is heavily promoting with Aegis, Progressive Peripherals & Software, NewTek, SV SoftwareVision, Associated Computer Services and Brown Wagh. All of these companies exhibited new Video software for the Amiga at COMDEX Spring.

ABACUS showed a number of products, at the Atari booth, including their very complete Atari ST library. Arnie Lee, President of ABACUS, also had a booth in another part of the show.

Navarone staffers showed their new scanner for the ST. For impact, Vince Martel scanned a $20.00 bill onto the Atari Mono Monitor screen. The effect was very startling, and with their outside corner location, a real stopper.

OMNICARD

Gordon Monnier and his group held forth with a batch of new Michtron software to be released shortly. OmniCard, a data base similar to HyperCard on the MAC was an eye catcher.

It runs very well on the ST. OmniCard should give HyperCard a run for its money in both price and features. Michtron's software is always reasonably priced and excellent quality.

Of course, WordPerfect was showing at the Atari booth as well as a large booth in the East Hall. Interest in this program was very high.

HiTech Advisors were there in force with the Musso's, Paul and Joyce, demoing updated versions of their point-of-sale programs with much interest on the part of dealers. Their Video Dealers software makes running a Video Rental Store a snap. All you need is the right location.

B.E.S.T. has one of the easiest accounting packages we've ever seen (we'll be reviewing it shortly) and, as usual, they had a crowd around their section of the booth amazed at the ease with which this program can be used, even by the neophyte.

Their contention that you don't have to be an accountant to run their accounting package is as near to the truth as we've seen anywhere. It's simple and very difficult to make an error with. The program makes all the decisions for you. A great product.

Illiad Software, with Dave Showalter in charge, demoed a number of products including Athena II, a Computer Aided Design program that lists for an amazing $29.95, CircuitMaker a simulation program that allows the design, construction and testing of an unlimited variety of digital circuits, at $79.95 and Teacher's Pet (they don't use the apostrophe) which tracks students and automatically recalculates after grade corrections, at $49.95.

Design Marketing Communications showed its Calamus Desktop Publishing program (available in June '88). At a list of $449.95, this is the most expensive ATARI program yet.

Designed to run on the Mega2 or 4, this program appears to be very powerful. There is a scaled down version for the 1040ST at $199.95.

Since it's difficult to determine the worth of a program under demo conditions, (we have never worked with Calamus) we'll have to run this one through it's paces during review to find out just what's different from the other DeskTop programs out there.

INTERSECT Software had two excellent programs up and running. Resolver, allows you to stop a program, get another program up for data or manipulation and then go back to the first program exactly where you left off. It will also save your place in a program bringing you right back to that point on boot up. At $49.95 this is a must program.

BUSINESS SOFTWARE ON THE ST

A second INTERSECT Software program is VideoMan, an accurate management program for Video sales and rental stores. This program covers everything including management reports. Because of the data requirements, VideoMan requires a hard drive but can be run from a 520ST.

An item outside the ATARI booth that impressed us was a READ/-WRITE CD device from MAXTOR, an OEM (Original Equipment Manufacturer) based in San Jose, CA. Started in 1982, MAXTOR now has over 3100 employees in the US and Asia.

MAXTOR has been a supplier to such companies as Apollo Computer,

(Cont'd on Pg. 4, Col. 1)

COMDEX SPRING '88...

(Cont'd from Pg. 3, Col. 3)

Data General, Digital Equipment Corporation, NCR, Texas Instrument and Unisys. Their main products up to now have been Winchester and conventional CD-ROM drives, with an 800 Megabyte Write Once/Read Many times (WORM) optical drive being the most popular.

According to Richard Green, CD Product Manager at MAXTOR, their read/write device will be available in in October '88 in a 5-1/4" size with 650 megabytes formatted and a 3-1/2" drive with 160 megabytes formatted. The disks are both removable and eraseable and are currently going in the $45.00 and $95.00 range. MAXTOR W/R drives are currently priced at $1900, plus or minus a dollar or two. The price per byte of storage is going down very rapidly.

Tandy, the first to announce a read/write CD device a few weeks ago, passed out a single sheet flyer at COMDEX (no product, specs or demo) on their TANDY THOR-CD (Tandy High-intensity Optical Recording - T-H-O-R, get it).

Although Maxtor has complete specs including SCSI (Small Computer System Interface) support and even reliability specifications, TANDY opted for an appeal to "organizations interested in participating in the development and production of the TANDY THOR-CD". Apparently they can't or don't want to do the development themselves.

Of course, TANDY's THOR-CD is targeted at $500, 1/4th the price of MAXTOR's, but TANDY's is two years away from seeing the light of day. The technology and approach are different, as well.

VERY HI RES GRAPHICS

Another company we've been following is HOWTEK out of Hudson, NH. HOWTEK makes high end (very high end) computer color input and output devices, flat bed color scanners and solid ink jet color printers. Their products produce very hi-resolution, approaching a conventional photograph.

At COMDEX Atlanta, HOWTEK exhibited a new scanner for 35mm film (negative or positive) that provided the highest resolution and color saturation we've ever seen. Of course we have champaign tastes and and a beer pocket, the scanner sells for $8000 and the complete system (you need it all) is over $100,000. Oh well, like everything else in the computer industry, prices will come down. But, in this case they have a long way to fall.

The laptop industry is at a standstill awaiting new screen technology. Back lit screens help but are not the real answer. If you have to work on one of these things all day, the screens will kill you.

There were plenty of clones, with a new one, DELL Computer, coming on strong. DELL (started by a group of college kids) is out of Fort Worth, TX and making a big splash. Everything is big in Texas.

Everyone, except IBM and Compaq, is suffering from the unfortunate decision by our government to embargo the Japanese chip industry for dumping. Never known for good timing, our Customs agency placed the embargo too late.

US chip makers had already shut down their lines and then Japanese product dried up. 256K and 1 meg chips are really short to the point that US computer manufacturers aren't stuffing their memory boards full in order to stretch the supply.

Predictions at COMDEX regarding the alleviation of this situation varied from this summer to early 1989. No one said it's going to happen yesterday. Our best guess is it will start to let up around September and we'll have a glut of chips by next Spring.

CHIPS SCARCE BEFORE

This is not the first time chips became scarce. Way back, the 64K chips were really tight for awhile and everyone was scrambling for product to keep their production lines running.

If you think going to COMDEX is work, you're dead right! Even the parties are tough on the feet with far fewer tables and chairs than the number of people invited. Monday night was the big Zenith party with plenty of interesting people to talk to. Zenith is pushing its laptop computer line and had the whole series on display and operating at the Marriott Marquis Hotel.

Tuesday night was the ATARI party mentioned above. ATARI was very considerate with plenty of places to sit and talk.

Wednesday night was the biggest party of them all (there's always one). Hayes (of modem fame) threw a monster affair in one of the Omni Hotel ballrooms (right across from the Convention Center).

The food was fantastic and there was plenty of it, even for the 1000 plus who showed up and danced after standing all day. The band was great and very professional, the music varied but danceable.

Being nonsmokers, we notice anyone who "lights up". If we don't see them right off, we smell them immediately. No smoking on the flights down and back were a god-send. Even at the show, very few people smoked, and they appeared to have a guilty look every time they took a puff. The world is changing.

COMDEX Fall (Nov. 14/18) in Las Vegas will be the 10th anniversary of COMDEX. The industry will be coming out of the chip problem and loaded for bear. ATARI will have 4000 sq.ft. of exhibit space to luxuriate in and make the statement that it's definitely in the US market and ready to take on all comers, on a level playing field.

Instead of "Wait 'til next year", we'll only have to wait 'til Fall. We'll be there! ▲

PARALLEL vs SERIAL on the ST

Aborted print commands have plagued us while using our laser printer, with as many as 5 tries to get a single page through.

Finally at COMDEX, Atlanta, we found someone had the same problem and by switching to parallel operation, they cleared it.

We couldn't wait to get home to find out if it worked. It did! Now we can do multiple pages on our Mega ST4 much faster than through the serial port, even at 9600 baud.

However, we can't do bit mapped graphics on our 1040ST using an LQ-1500 Epson printer (parallel output). We understand the YAMAHA sound chip provides the parallel output. Could this chip/s be marginal in both computers? We have two chips on order and will report the result as soon as the situation is resolved. ▲

NEWS AROUND

By Carole Terp

The first read-write CD ROM was announced at Comdex, Atlanta by Maxtor - cost about $1995 and shipping in October. Tandy's THOR-CD erasable CD ROM drive will be 2 to 3 years away. Drives and media suitable for playing and recording music will probably be available in 18-24 months and cost under $500. Optical Storage Solutions Inc., Concord, CA and Optical Media International, Los Gatos, CA are joining in an effort to establish standard file formats, etc. for erasable optical and WORM disks.

Toshiba and NEC are shipping read only CD ROMs this summer. Toshiba's will be under $1000, and NEC's is under $1200. Both are faster than Apple's $1,199 drive. The CD ROM version of Hypercard is now shipping, according to Hypercard author Bill Atkinson, and version 1.2 offers multiple access to read-only and no shared access to read/write media.

At COMDEX a number of flat screen monitors were displayed, and laptops were in abundance. Things are getting smaller and smaller. Paravant Computer Systems Inc., Melbourne, FL is selling a complete IBM-compatible computer the size of a videocassette. It costs $3,995 and can run most programs written for the IBM compatible machine. (Small doesn't mean inexpensive.) Its screen is 1/4 the size of a conventional monitor and the keyboard, being so small, has to be used like a calculator.

CYBER PROGRAMS

Saw Cyber Studio Cad and Paint programs in the Atari booth at COMDEX. What super programs! We can make things tumble, zoom and move around on the screen the same way the television stations do, at a much lower price. Antic is selling these in combination or separately.

ZOOMRACKS PATENTED

Quickview Systems Inc, Los Altos, CA has been issued patent No. 4,736,308 for Zoomracks' card-and-rack metaphor, how information is stored, accessed and displayed. The patent allows Quickview to license its card-and-rack metaphor to other companies interested in adding these capabilities to their own systems. Apple, we have their number if you're interested!

Sandia National Laboratories, Albuquerque NM is developing thallium for superconductor chips which should speed up commercial development of superconducting electronics.

Sterling Castle Software of Marina Del Rey, CA has designed Forget-Me-Not ($80 on IBM-compatibles) to pop up messages on the screen to remind the user of telephone calls, tasks, appointments. Make the computer work for you!

Microsoft is shipping its first OS/2 software, Multiplan 4.0. By the way, Microsoft has asked the court to split Apple's case against Microsoft into two parts - the first whether a 1985 license agreement between the two companies gave Microsoft certain rights to Apple's Macintosh interface for Windows and the second copyright infringment.

Hewlett-Packard, the other firm in Apple's lawsuit, has asked the court to dismiss one of Apple's law firms because they represent HP in other matters. The case gets messier and messier.

Borland's first OS/2 product is Paradox at a cost of $725 for a single-user and $995 for a Network Pack. Paradox for 80386 running DOS costs $895.

Everex will be making a complete line of IBM PS/2-compatible systems and is negotiating with IBM for a license.

Speaking of IBM-compatibles, remember Epson's QX-10 the first streamlined computer that runs Valdocs (the first integrated program)? Well, the sturdy QX-10 with its 5 expansion slots now has IBM compatible 640K board capability, including an EGA, VGA, Hercules adaptor board. Titan Technologies of Ann Arbor, MI is selling both of these boards.

Weitek Corp, Sunnyvale, CA has a Postscript processor chip that will make Postscript processing faster. Everything is getting faster.

DELL SELLS

Dell Computer is certainly kicking the prices down with its line of IBM-compatibles ($799 for an 8088 system to $7,399 for a 20MHz 386 system with the latest 32 bit architecture for MS OS/2 compatibility). They ship direct to the consumer.

Letraset USA, Paramus, NJ will have its typeface manipulation program Letra-Studio ready to ship this fall. A designer can reshape, resize and place characters precisely. LetraStudio supports the Pantone Patching System to specify colors on your screen and spot color separations. You can do anything with type fonts now. They also have a Letraset Electronic Type Library with more than 1,000 typefaces. The fonts output as EPSF (Encapsulated PostScript Format) to a printer. The first library version will ship with

(Cont'd on Pg. 10, Col. 3)

COMDEX SPRING '88

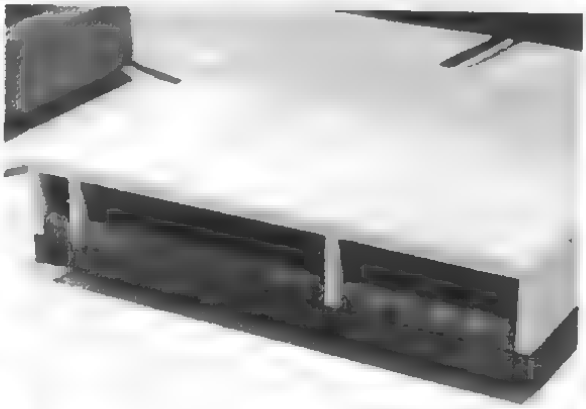
Robert Delisa of Progressive Peripherals demonstrates "Superbase Professional".

Shawn Fogle, President of SoftLogik Corp. shows "Publishing Partner Professional".

Stephen Yenney, Programmer&Pres., Jazzman Software, shows Don Terp"VideoMan" (sold by Intersect).

John Townsend demos CDROM.

John Pratt of Digital Vision stands next to his image on the computer screen illustrating "Computer Eyes".



Atari's CDROM drive due for shipment this summer.

Neil Harris, Director of Product Marketing, Atari Computer.

Gordon Monnier, President of Michtron, demos new program "OmniCard" including synthesized speech.

Roger D. Brown, Pres. of Navarone Industries tells about new scanner working with the Atari Computer.

Paul & Joyce Musso of HiTech Advisers inform us about their new Video Dealers software.

Green of Maxtor shows the 3-1/2" disk for the rite CDROM due to ship in October.

David Fletcher of ISD Marketing Inc. introduces us to "Dyna Cadd"

THE WRITER'S CRAFT
Post Production Treatments

By Don Terp

Have you ever noticed how easy it is to read some people's writing and how difficult to read others. They may both use the English language (or the Americanized version of it), they may both use correct punctuation and grammar, and yet one is much easier to read than the other.

Practice has something to do with flow, clarity and readability. Certainly anyone who writes all the time learns to be clear and readable.

Vocabulary is another factor. Some like to demonstrate a vast command of words by using unfamiliar ones in quantity, slowing the whole process down as the reader struggles to decipher phrases that should have been stated more simply.

Of course, the use of common words to put a point across is a favorable attribute. Comparisons with well known phrases makes association easier and ups the mental capacity to grasp the thought.

Some say that short sentences and short paragraphs help comprehension. That's somewhat debatable. Sentences with flow and paragraphs that don't backtrack a thought are clearly understood even if they're medium length.

One of the most profitable things a writer can do is to set his material (once written) aside for a few days before editing. It's amazing how much one can see once the original arrangement of words have left the brain.

Writers working against deadlines may not have the luxury of time to reread and edit their material. Perhaps it can be put in someone else's hands for the reread.

Someone with less knowledge of the subject being dealt with would be ideal. If there is something they don't understand, it's a sure bet the general public won't either.

We have written before about the fallacy of leaning on a spelling checker to catch all typos and creative spelling. The English language is chock-full of sound-a-likes (called homonyms) that pass right through a spelling check and leave uninitiated writers with egg on their collective faces.

They're and their, its and it's (one of our downfalls) come to mind quickly, as homonyms that are often, incorrectly interchanged. Homophones, close cousins of homonyms, also sound alike but have entirely different meanings. Two, too and to; sow, sew and so; air, ere and heir are examples of homophones. Read, reed and rede; rode, road and rowed are others.

If you would like to learn more about words (besides reading the dictionary), we recommend "The Writer's Art" by James J. (for Jackson) Kilpatrick ($8.95, Andrews, McMeel and Parker, Kansas City/New York). This Scrabble, VA resident has done more for the American version of English than anyone in modern times.

We hope your local newspaper carries his Sunday Column on writing, called interestingly, "The Writer's Craft". Jim enjoys words, enjoys playing with them and enjoys reading about mistakes made by writers who should know better, people who are paid to know and too often don't.

Kilpatrick also writes a column during the week on social commentary, but his Sunday efforts on writing are a prize and should always be clipped for future reference.

In writing, technical quality is assumed. When misspelled or incorrectly used words appear, the writer's credibility is damaged, readers question the research, the topic and the facts presented. The work is devalued, regardless of the time and effort that went into its production.

If we want readers to believe us and listen to what we have to say, it's important that we say it with technical writing excellence. This is a "given" in the writing world. ▲

THE NEW, NEW ATARI . . .

(Cont'd from Pg. 1, Col. 2)

products that are not in the current product line, except the CD-ROM player and that's due to ship in June.

Here was a good sign, one to add credibility and hush the detractors. Here was an opportunity to change the image.

As low profile as ATARI was at COMDEX Atlanta, there are already plans underway to make COMDEX Fall a great extravaganza. ATARI will be taking take over, exclusively, the 200 X 200 foot "Gold Room" at the Las Vegas Convention Center during COMDEX Fall (Nov. 14th/18th). This room in the main convention center, near the entrance/exit to the Las Vegas Hilton and just a few feet from the main entrance will give ATARI an opportunity to make a loud statement regarding its intentions in the computer market.

In the fall, the chip shortage should be alleviated or a solution within sight and ATARI will be ready to go again. Both developers and Atari will gain enormously from the prestige of having a large room for their exclusive use.

The "Gold Room" is over 4000 square feet of space, without question one of the largest single exhibit sites at COMDEX, as large or larger than IBM, Compac, Aldus, Kodak or, for that matter, any exhibitors, even those with spaces in two different locations.

The foot traffic around the "Gold Room" is always very high due to exhibitors both in the Hilton and Convention Center pulling attendees through this corridor. In the past, we have actually seen jam-ups during peak times.

So, while ATARI has chosen to make a low profile statement during COMDEX Atlanta, despite a severe product shortage due to the DRAM situation, it's working toward November and a smash success in Las Vegas.

This will be the NEW, NEW ATARI in action. Hang on to your hats! ▲

DESKTOP PUBLISHING
Getting Into Print

By Don Terp

You have just written a book and now you're trying to get it into print so you can sell it and make some money. Sorry to say, your struggle has just begun.

One of the first thoughts is the Vanity Publishers. They advertise in writers' magazines offering to publish and distribute your book for a nominal fee.

Unlike a conventional publisher who pays you an advance against royalties, vanity publishers are paid up front and you get all the copies to do with what you want.

People of means like to say they've been published and don't mind paying for the prestige, hence the name vanity press. There is much hidden in their contracts, and their performance is almost always less than what you could do yourself with a little effort.

In addition, most publishers won't consider a vanity press publication as a credit on your list of books published. They'll consider a self-published book first.

We have recommended a number of books (all self-published) in past issues of ST BUSINESS and continue to recommend those same publications as a road or path toward self publishing. They are "must read" for anyone just starting out.

Another traumatic experience will be the selection of a printer to print and bind your book. We have encountered a definite resistance to handling typesetting that isn't done "in house".

One of the ways this is visible is the outlandish prices some printers quote for printing. They figure that if they aren't going to get the typesetting, they're going to charge for it anyway.

By quoting an "over all" price, they can bury their mistakes and errors and still make money. This is all at the expense of the person purchasing the printing.

Insist on an itemized quotation. One that shows the price of the paper, negatives, plate making, press time, folding, binding and packaging as separate items. This is no trouble, since the printer must create this data for his own quotation.

A good idea is to have a standard quote form so you can compare figures and find out exactly where you want to give and where you don't. After all, you could be spending upward of $6000/$10,000 for a good printing job. You have a right to know exactly what you're paying for.

One book that can be a great help in this unfamiliar area is the "Directory of Book, Catalog and Magazine Printers" by John Kremer ($15.00/Ad-Lib Publications, 51 N Fifth Street, Fairfield, IA 52556-1102.

John has expanded this book in the 4th edition and included tips on selecting printers, printer's terms and much more.

Before you chose a local printer who may be able to do your book but is not set up to do it inexpensively, read John Kremer and learn how to save money. His information on RFQ's (Requests for Quotations) alone is worth the price of the book.

Tell him you saw it in ST BUSINESS! ▲

BOOK REVIEW

"The Atari ST Book - Tips, Instructions, Secrets and Hints for the 520, 1040, and Mega STs"

By Bob Jones

Ralph C. Turner
P.O. Box 1822-3
Fairfield, IA 52556
$16.95 - direct from author
(add a dollar or two for shipping)

If you're anything like us, you have a large collection of books about your ST computer as well as books about other computers, operating systems, software and hardware. Our collection is very, very large and grows every time we get to a book store. Each book purchased seems to have additional information, a new approach or treatment of a topic.

Along comes a book that is different, different in that it's very useful on every level, from the rank beginner to the most advanced ST owner. The "Atari ST Book" doesn't explain the C language, BASIC or how to use WordPerfect. What it does do very well is provide information in the form of "tips and tricks" on using the TOS operating system, GEM, changing control codes using a word processor, printers and many other [illegible]

You know all that, you say. We do too, but it's much easier to look it up in Frank's handy book than to sit down and try to figure out just what to do each time.

In the works for over a year, Turner personally tested and used each procedure he recommends. The section on the use and applications of AUTO folders is very well done and particularly easy to follow.

With 17 chapters and 160 pages, this is no pamphlet, but a serious work on the ST. It's a must have for anyone just starting out on this machine.

The warnings about Delete, Copying Between Windows, connecting printers and shortcomings in the GEM Item Selector are alone well worth the price of the book. There are many more well researched and documented, practical hints to make your ST operation more enjoyable and productive.

Much of the text is a series of

(Cont'd on Pg. 11, Col. 1)

SMALL BUSINESS CORNER

Communicating with your Customers

By Eric Neilsen

When is the last time a supplier called and thanked you for an order? When is the last you heard from the top man in a company you do a lot of business with?

We don't let people know we appreciate their business. The only communication is the monthly bill with a few inserts asking you to buy more.

In one case recently, we received a bill with no balance (we didn't owe them any) and a computerized note recommending that we buy something real soon. We hadn't bought anything in that store for some time because they weren't meeting their competition.

Your business changes over time. You drop old products and services and add new ones. Infrequent customers never find out about these changes until or unless you tell them.

How much would it cost to run a monthly or bimonthly news letter chatting about the goings and comings, new products and new personnel? Not a lot of money, not even a lot of time.

If it just broke even (expenses vs sales) you'd be ahead. All experts we talk with say the projections are much better than that. It's almost impossible to loose money on a good solid customer contact unless it's completely out of hand expensewise. Everyone likes to believe they're individuals, someone whom others take an interest in. Business has gotten so impersonal with all this digitized, computerized and sanitized management that everyone's afraid to say hello or thank you.

If you have an account you haven't heard from for some time, get on the horn and talk about it. There may be a problem with your business or his. You may be able to straighten it out in a brief conversation. At least, you'll have a better idea of what's going on in your customer's mind.

Above all, don't send a dummy bill with a note suggesting that the reluctant customer get off his duff and give you some business. He may do just that.

Above all, don't relegate the job of contacting the customer to a subordinate, secretary or the telephone operator. If you're running a well trimmed ship, you should have the time to take care of important customer contacts yourself.

Perhaps your salesman is having a personality conflict with a customer, everyone doesn't get along with everyone else. A quick phone call will put you in the know and give you facts to aid in the decision making process.

If you're running a multilayered organization, with distribution so masked that you'd have no reason or excuse to contact the end customer directly, a surprise visit to your distributor may be in order. The surprise may be on you when you find out just what's going on.

You may think business is on the increase while the competition is swamping your ship with a 100/200 or 400% order increase. Your distributor should have known that and told you. He may not have had his ear to the ground or his salesmen aren't really working.

One way to find out, pay a visit. Let 'em know who you are and what you expect. Unless you tell them, they'll never know.

Customer contact is a very important part of any business. It separates the businesses building a future from those who are content with living in the present.

Thank them for the order. You'll stand out in the crowd. ▲

NEWS AROUND . . .

(Cont'd from Pg. 5, Col. 3)

100 fonts. The rest will probably ship during the first quarter 1989. Price has not yet been determined.

Turbosoft, Cleveland, OH is selling a program called Overdrive that has a set of 50 legal documents and macros for WordPerfect. Cost? $99.95.

Apple is shipping its CD ROM player for the Macintosh at $1,199, and is shipping an 80-megabyte hard disk subsystem for $1,249. We saw Apple's Macintosh II locally and were very impressed by their high resolution monitor. It was just as sharp in color as in monochrome. IBM programs can be run on the II and copied right into Apple programs. Apple still chose not to be in COMDEX/Spring '88 show in Atlanta.

Vittorio Cassoni (former AT&T Data Systems Group President, now with Olivetti) keynote speaker at COMDEX, Atlanta warned that Apple will suffer for maintaining a proprietary operating system. Cassoni stated that open systems based on UNIX are the only viable computers of the future.

Due to dollar-yen exchange, Hewlett-Packard has increased price of the Laserjet Series II Printer. When will our politicians quit meddling?

400 DOTS PER INCH

Panasonic's flatbed scanner can scan at up to 400 dpi (dots per inch) and has suggested price of $1,899. Scanner can be obtained from Panasonic Industrial Co, Secaucus, NJ.

The Macintosh users will be able to share files on UNIX machines from 40 different vendors this summer. File server software called uShare will make it happen.

Users of PowerPoint by Microsoft can link directly to Genigraphics' slide bureaus. PowerPoint 2.0 for the Mac is $395.

Ashton-Tate is shipping FullWrite Professional, its word processing program acquired from Ann Arbor Softworks Inc. It combines, outlining, word processing, drawing and page layout at $395 for the Mac.

Rob Shostak, who along with Richard Schwartz started Ansa and created Paradox now distributed by Borland, has left Borland International to start a new software development company. He will consult for Borland. ▲

A REVIEW

A-CALC PRIME and A-CHART

BOOK REVIEW . . .
(Cont'd from Pg. 9, Col. 3)

questions with answers being a description of the hint or tip, an easy approach that does not lecture to the reader or make author Frank Turner appear to be an expert beyond the scope of the material.

Much is made here of PD (Public Domain) software, particularly the utilities available on many ST bulletin boards and sold in the form of disks by User Groups and independent retailers.

The author recommends these utilities very highly although he provides few direct sources, allowing the reader to fend for himself in this area. Recommending GENI, THE SOURCE and User Group libraries doesn't solve the problem for the fellow out in the boonies who has no modem or the inclination to get into very expensive electronic bulletin board activity to retrieve PD software. A few recommended, well researched, retailers would have been an asset here.

He does, however, provide a number of resources including a very extensive list of ST publications (including ST BUSINESS) and a number of suppliers (software houses) mentioned in the book. The INDEX, too, is well done and clearly referenced, a must for any practical use.

Working on Book #2, Turner asks each and every ST owner to write to him (address above) with a tip, hint or instruction to be include in the next volume. We must assume that he will give credit to the first receipt of any material.

This is not a rewrite of the Atari Manuals, the Developer's Manuals or any other material. It's made up of tricks and tips that Ralph has found in using his ST and decided to make available to the ST community.

Recommended! Can't wait for Volume #2. ▲

ANTIC Software
544 Second Street
San Francisco, CA 94107
(415) 957-0886
A-CALC PRIME -- $39.95
A CHART -- $19.95
Not Copy Protected - Don't Pirate

For some reason, only a psychologist could explain, people who buy software must always have the best, latest, most expensive and most complicated. Never mind that they'll become frustrated, spend needless learning time and never really make use of all those "pro" features, they must have it because their egoes demand it.

A-CALC PRIME is not Lotus 1-2-3, but then neither is the price. Here is a topnotch program that does 85% to 90% of what Lotus does for very little money.

It's not going to tax your brain, your prestige or your qualifications as a bean counter (accountant) either. A-CALC PRIME will get you there and back, that's all you can expect of any software.

First off, A CALC PRIME is professional software, it's as good as any spreadsheet we've seen. In fact, as an ST owner, you'll find it better than anything else you can buy.

It is written for the GEM desktop (the GEM interface is well worth 20 times the price). It uses all the GEM features and adds a few of its own. In fact, it's the first GEM based spreadsheet we've seen.

Note: If you buy this program, be sure to print out the README. This one really does update the manual with some new features. It also corrects a number of errors in the manual. Read it before you become discombobulated.

A-CALC PRIME is actually an upgrade of the original A CALC 1.0, written in merry ole England and sold there under the KUMA Computers label. It offers many bells and whistles only available on high priced software along with some features of its own.

If you're familiar with A-CALC 1.0, you can now lock columns and rows, execute conditional expressions, use Macros and labels plus run your sheet out to 8192 rows by 256 columns. Trig functions, sideways printing (a must for long sheets), password-protected cells, full mouse control, and function referencing with keyboard alternates are also included in A-CALC PRIME. Ten clipboards and very flexible printer formatting are also new.

If you've never used a spreadsheet or have little inclination toward math, you might find the installation and startup of A-CALC PRIME a little taxing. Actually three books on spreadsheets are recommended on pages 2 and 3 of the manual.

We recommend all three regardless of your level of expertise. There are tricks you could never learn or discover in a million years. Mastering spreadsheet manipulation gives one power that few can relate to unless they've had the experience. Somehow, controlling all those numbers, formulas and cells is mindboggling.

Those of you who have been reading us for some time, know that we move data from one spreadsheet to another. In our case it's the YTD (Year To Date) column in one month's sheet to the Previous Month column in the next. Since, under GEM, you can open and transfer data between up to four windows, what has in the past been a problem with many errors now becomes a snap.

If you've never used a spreadsheet under the GEM Desktop, give A-CALC PRIME a try. You'll be very surprised and happy once you stop thinking about line commands, back slash commands and function keys. Oh, they're there if you really must. Bets are, you'll be using the mouse

(Cont'd on Pg. 12, Col. 2)

REVIEW . . .
(Cont'd from Pg. 11, Col. 3)

within the first few hours.

A-CHART A GRAPH PROGRAM WITH MANY FEATURES

Graphs and charts are always easier to grasp than narrative. There is something about the visual concept of a graph that people comprehend quickly and accurately. A-CHART produces those kinds of visuals.

Inexpensive, easy to use and very professional in results, this full featured program is a steal at its current price.

It can use the output from A-CALC PRIME, any other DIF (Data Interchange Format) file, or data can be created within A-CHART itself to produce graphs. Just like A-CALC PRIME, A-CHART is a GEM featured program that is easy to use and produces top notch results.

A-CHART can draw Line, Scatter and Area graphs, 3D bar, Stacked bar, Vertical bar, Overlaid bar, Horizontal bar and Pie charts. All graphs and charts are capable of showing multiple data sets except pie charts. Only one data set can be displayed in Pie chart.

Although up to 80 data sets are allowed across all the graph styles, only 20 can be displayed on any one graph. 30,000 data points can be plotted using a 520ST, with 100,000 points available on a 1040. Need more, get a Mega ST4.

Scales are only available on the left (Y) and bottom (X) axes (axis of ordinates and abscissae). No multiple ordinates can be accommodated (you can't have left and right Y axes).

Different line and fill patterns can be selected to separate data and make a graph or chart more readable. You can also switch back and forth between types of charts as A-CHART redraws the selected graph or chart (remember, only one set of data for a Pie chart).

In addition to conventional charting, A-CHART has a STATISTICS MENU that can produce stats for a given data set. You may not need to know the Sigma Y Squared value or the 1st and 3rd Quartile value, but if you do, A-CHART's Statistics Menu can provide that and much more.

To anyone into statistics this part of the program is very valuable, since all of the stats are generated on a table in a GEM window. To do this same math with a hand calculator would take a month of Sundays.

If you need graphs, charts or stats, you need A-CHART. At $19.95 it's a super bargain that conceals its value and quality. Every business should have one! ▲

Desktop-Publishing:Pseudo Publishing

by Dan Fruchey

What do you call a program that's similar to desktop publishing but isn't desktop publishing? How do you describe a product that's similar to a wordprocessor but isn't just awordprocessor?

Confused? If so, you're not alone. In recent months we've seen the introduction of a series of ST applications that don't fit any current software category. How do you classify a program that uses multiple fonts and graphics like a publishing program, yet still allows spell checking, mail merge, and ASCII file creation like a wordprocessor?

Currently there is no official label used to classify these programs. As the products become more popular I'm sure some bright individual will coin a name for these programs that we can all use and understand. Are they turbo charged wordprocessors or publishing programs with most of their guts ripped out?

To reduce confusion I've labelled these products "Pseudo Publishing" programs. This arbitrary label may be considered a misnomer by some individuals but it will suffice for our purposes. This temporary label may also explain why these programs are mentioned here in a desktop publishing column.

There are three programs that fit in this pseudo category of pseudo software: **1st Word Plus** from GST Holdings, **Microsoft Write** from Microsoft and Atari, and **Word Up** from Neocept, Inc. In this article we'll explore the differences between Pseudo Publishing, wordprocessing, and desktop publishing. We'll also compare the three Pseudo Publishing programs that are currently available.

Why Pseudo Publishing Software?

The ST has desperately needed an intermediate category of software that bridges the gap between publishing and wordprocessing. Some users don't want to endure the difficulties of page layout just to add a picture to a document. Other users need wordprocessing functions that are not available using desktop publishing software.

Since I began testing these programs I have found myself using them as often as my wordprocessor or desktop publishing software. Pseudo Publishing was not created to compete with desktop publishing. It offers some of the most attractive aspects of desktop publishing to the wordprocessing market.

Pseudo Publishing and Wordprocessing

On the surface these programs resemble a wordprocessor with multiple fonts and graphics. Big deal. However if you spend time investigating Pseudo Publishing software I think you'll be impressed with its capabilities.

Just like a wordprocessor you can begin typing immediately. The normal style and formatting options are all present, it saves formatted and ASCII text, spell checking is available, and you can obtain a hardcopy without possessing a college degree. In short, Pseudo Publishing can do whatever a wordprocessor can.

Pseudo Publishing will also allow you to change fonts, resize text on screen, access fonts that are not supported by your printer, and add diagrams and pictures to documents to give them a more professional look. You can adjust spacing, justification, hyphenation, leading, and gutters to increase the readability of your documents. Glossaries, a clipboard, and the ST's extended character set are readily available.

Drawbacks of Pseudo Publishing software include slower scrolling speeds while scanning text, greater RAM requirements when fonts and graphics are added, slower printing speeds, and a range of functions that could complicate work if you just need a simple letter.

If you plan on using an occasional graphic for a diagram or want to add different text sizes for simple handouts and newsletters then I recommend you buy one of these programs. If you plan on performing intensive graphic and text manipulations on a regular basis buy desktop publishing software.

Pseudo Publishing and DTP

Many of the options you have come to need and desire in desktop publishing software are also available with Pseudo Publishing. Fonts and graphics used by publishing software can also be used by Pseudo Publishing software. Multi-page layouts, a master page option, versatile page sizing, additional style options (outline, double underline, strike through), and a higher density of printer output are all supported.

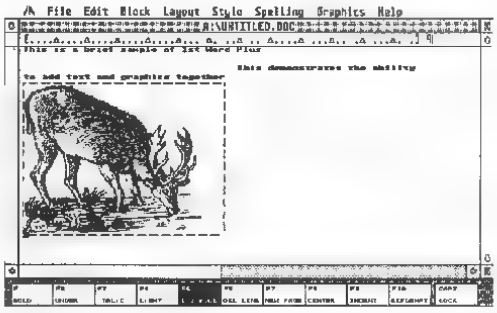
Pseudo Publishing software normally prints a page faster than desktop publishing software, requires less time to format page layouts, and demands fewer contortions to reformat text. It is superior in its capabilities to spell check text, mail merge, create ASCII output files, create footnotes, and search and replace text along with many other wordprocessing functions.

These advantages over publishing software may seem persuasive but there are just as many disadvantages. Pseudo Publishing software can't create graphics - it can only import them. Layout options are restricted while desktop publishing allows free form page layout. A maximum of two layers of text and graphics are allowed by Pseudo Publishing software while in DTP the number of layers of text and graphics are virtually endless. Graphics can't be rotated, distorted, mirrored, edited, or copied. There are no zoom modes, no grids, no alignment commands, and only one page can be viewed at a time. Fewer graphics can be used on a page, pictures won't be as large, and text can't be overlapped.

If you need to create or manipulate graphics from within your program desktop publishing software is a necessity. If you plan on creating text heavy documents such as books, manuals, or flyers that require few graphics then Pseudo Publishing software is a smart choice.

Pseudo Publishing Software

All Pseudo Publishing programs use basic wordprocessing functions. To save time and space the listing below only mentions options that make these programs unique.



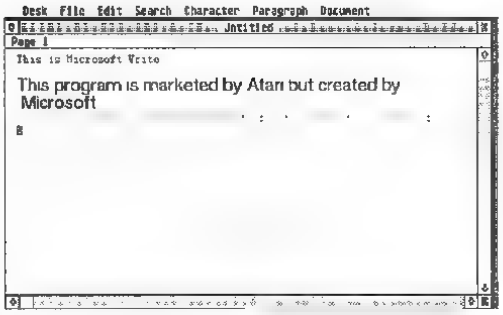
1st Word Plus - This remake of GST Holdings popular wordprocessor, **1st Word**, will be familiar to many users. 1st Word Plus allows importation of image (.IMG) graphics that are up to a full page in size. A utility included with the program also allows creation of image graphics from Degas, NEO, and Doodle picture formats. Graphics can be moved on the page as desired once they are loaded. You can even type over graphics to fill banners and caption graphics.

This program only uses the font set built into your printer. You can create headers and titles with a paint program and then merge them into **1st Word Plus** as graphics to overcome this shortcoming. All the documents you have created using **1st Word** and **Word Writer** can be imported with formatting commands intact and then graphics can be added.

The program also includes hyphenation, a dictionary, footnote functions, a mail merge utility, and a range of formatting commands that far exceeds the original **1st Word**. This program is well documented and virtually bomb free. There is only one bug that I am aware of, the program doesn't allow you to input document headers. This minor bug should be fixed by the time you read this article.

1st Word Plus was the first Pseudo Publishing program available on the ST and is easily the most user friendly. The program works with a 1 meg system without problems - something the other programs may have difficulty handling. The only drawback to using **1st Word Plus** I experienced is its inability to resize graphics once they are loaded. Still this is a minor detraction when compared to the many problems associated with the other programs.

I'm afraid I was unable to locate a U.S. address for GST Holdings. Contact your dealer for further information. **1st Word Plus** retails for $100.



Microsoft Write - This much anticipated port from the MacIntosh computer was contracted from Microsoft by Atari. It uses a wide variety of GDOS fonts and font sizes. Swiss, Dutch, and Typewriter fonts are included with the program and you can create or buy additional fonts as well. The program does not allow picture importation. **Microsoft Write** is the only Pseudo Publishing program that allows you to save documents as GEM metafiles and then export the formatted documents for use by programs that understand the .GEM file format such as Easy Draw.

The program allows you to use predefined glossaries of words, phrases, and formatting commands to speed document creation. **Microsoft Write** includes a clipboard for saving text, footnoting, a wide range of tab and page measurement methods, mail merge, adjustable gutters, multi-column output, auto page reformatting, and makes excellent use of keyboard equivalents for drop down menu commands. The print option allows you to print pages using your printers built in character set or the GDOS graphic fonts.

I have very mixed feelings about this program. A great deal of thought was put into program creation but the current version is buggy. The programmers overlooked ST abilities in porting this program from the MacIntosh. One annoying and detrimental oversight won't allow you to use the Delete key to edit text. There is no Delete key on the Mac (you use backspace instead) but there is no excuse for this oversight on the ST version. Hitting the Return key while a column of text is highlighted deletes the highlighted text, the list goes on.

The program has repeatedly bombed on me for no apparent reason and I loose my entire document. When I reboot the program I have to redefine the page size as page options reset themselves to 0" by 0" every time for no apparent reason. Scrolling is extremely slow and centering is often offset when using larger font sizes. Microsoft has washed their hands of the product and now Atari must fix their errors and oversights.

I am optimistic about the introduction of a fixed version of this program. **Microsoft Write** possesses remarkable potential but lacks true usefulness in the current version. I have repeatedly questioned Atari about upgrades online with little or no response. One comment made by an Atari exec on GEnie was that **Microsoft Write** was released too soon - I heartily agree.

For more information on **Microsoft Write** contact: Atari Corporation, 1196 Borregas Avenue, Sunnyvale, CA 94086, (408)745-

2367. Write retails for $129.95.

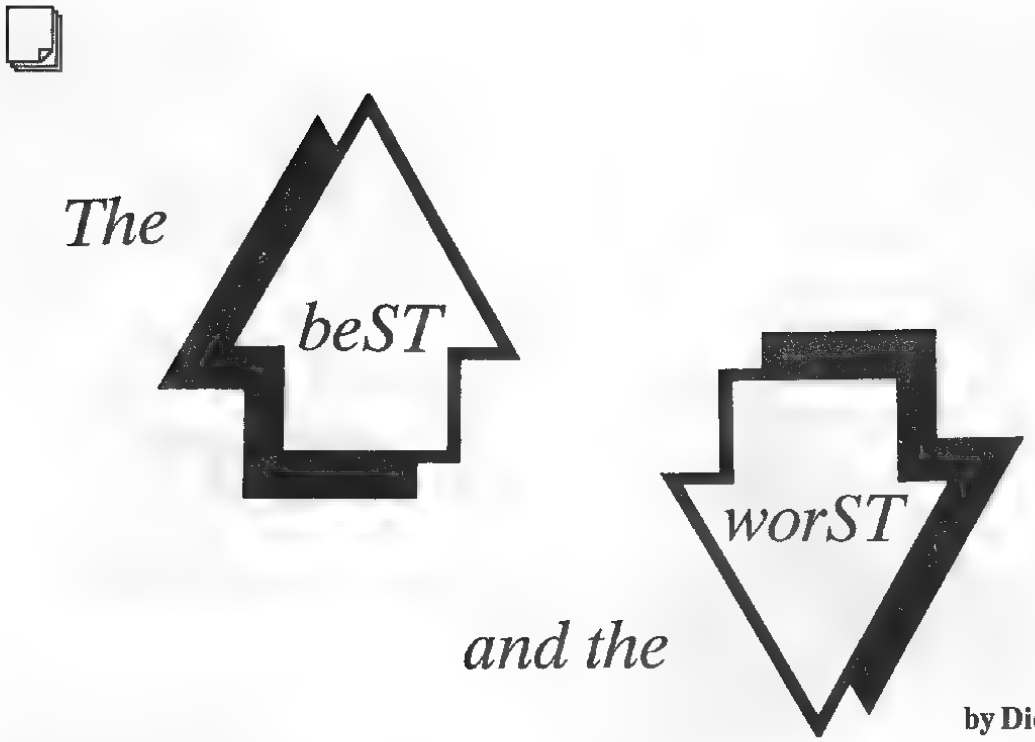
Word Up - Neocept, Inc. has recently introduced the most versatile Pseudo Publishing program available. It allows importation of image (.IMG), Degas (.PI1-3), and NEO graphics and the graphics can be sized from within the program. **Word Up** also uses multiple fonts and includes the same GDOS Swiss, Dutch, and Typewriter fonts used by **Microsoft Write**. While most programs limit the number of GDOS fonts available, **Word Up** allows a variety subject only to your RAM capabilities.

It includes a master page option, a clipboard, glossaries, mail merge, multi-column output, adjustable leading, variable justifications, configurable page layouts, an adjustable super/subscript option, and footnotes. **Word Up** makes use of several additional text style options including outline, double underline, broken underline, and strike through. It makes excellent use of keyboard equivalents to drop down menu commands and accesses the extended character set it a pain free way.

The program can be easily configured to take advantage of your personal style, font, and layout preferences. **Word Up** automatically reformats text, merges information from most databases, and a whole lot more.

While I feel this is the most versatile of the three programs it is certainly not the most user friendly. The program will run as is, but plan on spending some time to configure the many options to your personal preferences. Except for pictures, **Word Up** is RAM resident; this means printout occurs much quicker but more RAM is needed for serious work. The program will run on a 520 ST but don't anticipate using accessories or more than one or two graphics.

Continued on page 48 ==>



by Dick Biow

How to Succeed in Programming and Publishing

Can you write a popular software program if you've never gotten a degree in computer science? Can you start a successful business devoted to marketing that one program? Can you maintain that business if you've never studied business administration? Can you get it rolling with less than $25,000 front money?

Would I be asking those questions if we didn't both know the answer to all of them is, "Yes?"

What you've just read is what the advertising types call, "a grabber," a device to seize your interest and hold it long enough to make a pitch. The grabber has grabbed you, or you wouldn't have read this far. The pitch is that the one-man, get-up-and-go enterprise is alive and well in the United States. The proof is Wayne Buckholdt.

Wayne Who?

Wayne Buckholdt is the author of **Turbo ST**, the "software blitter" we told you about in the March issue of **ST Applications.**

Turbo ST is the program that speeds up GEM programs, usually more than Atari's blitter does. By now you've seen **Turbo ST** reviewed everywhere, usually in a very complimentary fashion, and you no doubt have it on your system disk. If you haven't seen or heard of **Turbo ST**, you must have been in a submarine for the last 4 months, lurking under the polar ice cap, studying the migration habits of seals from below. While you were doing that Wayne's company, SofTrek, went solidly into the black.

By the end of April, Wayne had sold 1300 copies of **Turbo ST** taking him past his financial break-even point. It wasn't an easy glide by any means, but Wayne is glad he made the decision to quit a secure job and jump out on his own.

Wayne, How?

We decided to pin Wayne down on his method, what mistakes he thinks he's made, and what it takes to become an entrepreneur in the ST software field. Generally, we tried to ask questions whose answers would be useful to the rest of us who might someday be tempted to tear the ST software market apart with a radical new program. We homed in on Wayne's specialty: the writing of extremely fast assembly code. Here's what he had to say:

The College Degree: Help or Hindrance?

"The absence of a college-degree might actually have been a stimulus in my particular case. Colleges concentrate, as they should, on teaching students to conform to conventional standards, emphasizing solid program structure and practical algorithms. Graduates

can be confident that they can program effectively. On the other hand, personality differences can influence a programmer's educational needs.

"For instance, though I didn't graduate, I have the practical equivalent of a computer science degree, since the courses I lack are all outside my major. There are two main reasons I didn't go the entire college route. First, I wasn't as disciplined as I might have been, and I had a difficult time forcing myself to conform to what was being taught. I liked to create alternate methods to solve assigned problems, to do things differently than we were shown, and so I often found myself studying what wasn't going to be on the test. Second, I was working full time by the beginning of my senior year. This certainly made college attendance more difficult, especially on my last job, where I was traveling out of town a lot.

The Creative Edge

"I think that the absence of a formal degree might have stimulated my desire to write superior programs. I'm out to prove a point -- that people don't always need college degrees to write excellent or fast code. So many companies withhold promotions from highly competent people, simply because they lack degrees. Degrees are measures of competence -- no question -- but the ultimate test of an individual's competence should be his performance in his job. Executives should evaluate employees with more flexibility than they frequently show.

"To overcome the handicap of not having a degree, I set out to compensate through extensive reading in the computer science and digital electronic fields -- always looking for ideas, fresh concepts, better approaches to old problems.

The High-Speed Edge

"Here's where my non-conformity just might have paid off. Schools in general teach you to 'modularize' your code, breaking each problem into many small pieces. This technique can be useful in keeping a program well-organized and easier to follow. On the other hand, it slows the program down, because of the overhead -- the extra operations involved in calling up procedures and passing parameters between them.

"In fact, I've seen many cases of one line subroutines, where simply doing the actual work as part of the main routine would involve much less time than calling the subroutine. Schools also teach you to write "general procedures" which handle a broad range of data. For instance they will tell you that a sort routine should sort any number of items, including zero, and not be limited to --say-- ten. This increases the sort routine's overall utility, but it often slows a program down, because general procedures are almost never as fast as specific-case procedures. For example, a routine that clips only horizontal lines is much faster than a routine which clips lines at all angles. ("Clipping" a line means keeping a line from being drawn outside a specific rectangle or area). So, if you are writing a program that draws a lot of horizontal lines, you may want to break out horizontal clipping as a specific routine.

Commitment to Fast Code

"Many software authors start writing "spaghetti code," where they jump back and forth over all the code, thinking they are saving time by not writing structured code. (In BASIC, this would be the use of GOTO instead of structures like FOR NEXT, DO WHILE, REPEAT UNTIL, etc.). Initially, one should write structured code that works properly. This step forces you to understand the problem better. Then go back, and consider how to optimize the code.

"This isn't just a matter of mechanically following one particular strategy or approach. You really have to believe that you can write fast code, and then develop a desire, almost an obsession, to match that belief. Pour over your code, looking for new ways to solve the same problem. At the same time broaden your knowledge, by reading books about other computers, languages, or specialties. Also, subscribe to some of the technical or popular journals to keep current, because the field is changing so fast. And last, break the rules if you have to; don't be restricted to what you've been taught.?

The Business Side

As you might expect, the individuality Wayne shows in his programming is reflected in the way he started up and now runs his business. Wayne saved $5000, borrowed $15,000 and went into business without the aid of venture capital. He brought in Bob Hanson, who receives a percentage of gross sales, to market the program. He employs someone to help answer his phone. And that's it!

That means Wayne wrote the program, distributed beta copies, corrected it, selected his own logo, and followed through with all packaging activities. He corrected Turbo ST to cure problems early users called him about. He made his own arrangements with European distributors. He's delighted that everything is going so well, of course, but he's real tired!

Observing his operation from its inception, I'd say that much of the secret of Wayne's success is to be found in his enthusiasm and drive. I've seen him force himself to just this side of that point where a positive response to self-induced pressure might give way to what the doctors call, "stress." For example, when he let me do the initial announcement of Turbo ST, his program was not completed. He knew that the date the announcement would be printed was his deadline for getting Turbo ST ready to ship -- including final programming, editing the documentation, package designing, and shrink-wrapping. This forced him to move, and he made it just in time.

But the point is, he made it, and now he's looking for new fields to conquer. For example, he knows he can write another "Turbo" that can speed up the Atari ST's

graphics the way his present program speeds text. Will we ever see it? That depends on the sales volume of Turbo ST. Notice, please, I said sales volume of Turbo ST, not popularity, which is enormous.

That's right, piracy will determine the future here. The amount of piracy of Turbo ST will determine whether Wayne's next program is for the ST market or the PC market, where MSDOS's huge installed base moderates the deleterious effect of piracy.

Sure, I know you wouldn't pirate software, dear reader. But if you have any friends who are using informal issues of **Turbo ST**-- any undocumented users, shall we say -- how about asking them to pay their tollaned pull their own weight. We'll all be winners if they do, right?

Subject Change - Why the ST Isn't Selling

According to the folks who keep track of these kinds of statistics, about 40-50 thousand ST's have been sold in the United States over the last 3 years. This figure hardly matches the expectations of those who predicted that the ST would become "the Commodore 64 of the 32-bit world" -- which is to say, it would sell by the millions.

One reason that ST sales have failed to meet Atari's goals is that the company never has convinced American computer buyers that the Atari ST provides "power without the price," as Atari boss, Jack Trammiel boasts. Oh, the machine fulfills that boast, all right. But Atari never bothered to prove it. Instead, they permitted the public to perceive the ST as seriously overpriced -- and they are still failing to deal with that misconception.

Pricing And Competition

The Atari ST's main competitor isn't the Apple, or the Mac or the Amiga: it's the PC-XT clone. The PC-XT clone is the major competitor of every computer that's near its price range. It is the first machine anyone thinks of if his budget is $1600 dollars or less. How do the Atari and clone prices compare? Well, you can pick up an Atari 1040 ST with one floppy drive and a monochrome screen, here in Orlando, Florida, for about $900. At first glance, that price looks highly competitive. In comparison, The PC Factory, Inc., in nearby Altamonte Springs, will charge $980 for an XT compatible with a mono screen, a floppy drive, MSDOS, and a lot less memory than what the ST delivers. (Yeah, yeah, I know about mail-order discounts for both machines, but discounts don't change the ratios significantly.)

Atari's problem is that few serious computer users and no business users have any interest in "entry-level" computer prices. Practical users shop for computers with hard drives, even if they intend to postpone the purchase of a hard drive until a later date. It is precisely here that the "Power without the price," argument falls apart: with a 20-meg hard drive, the price of the XT compatible rises to only $1210 dollars, while the 1040ST, equipped for example with a Supra 20-meg hard drive,costs $1599?

In other words, the customer is being asked to pay a premium of nearly $400 for the privilege of owning an Atari ST. Since Atari hard drives are enormously more expensive than PC drives, the hard-drive user concludes that he will be paying for "price without the power," if he opts for the Atari ST. And, in a price-sensitive market like the Atari's, a $300-$400 price gap is decisive -- unless it can be shown that the more expensive machine provides an advantage that clearly justifies its higher cost. This is something that Atari has failed to do. It can be done, though!

The Atari's Hard Drive is Faster

A 20-meg hard drive for the Atari ST, such as the Supra, is 3 to 3.5 times as fast as a 20-meg drive on an XT clone, in terms of effective through put into the computer system. With a 30-megdrive as the standard, the Supra moves data at 3-4 times the speed of an XT's hard disk.

That's what the extra $300 buys you -- speed! After all, speed is the main motive for buying a hard drive. Don't let anybody tell you it's just the larger magnetic memory he's after? If all he wanted was memory size, he'd be shopping for a tape drive.

Atari salesman's pitch to a reluctant, price-conscious buyer: "If all you want is capacity, why don't you buy a tape drive and hook it up to a Commodore 64? You can save all kinds of money that way, and 90-minute tapes have an enormous capacity. What? You say tape's too slow and capacity is useless to you without speed? Well, if you really want speed, why not get it? Why compromise with an XT compatible? Just consider how much time a 3X speed increase in disk access can free for you in a month's use of your machine! You agree? Fine! Sign right here!

And there you are, thinking, "Hey, I never knew all that!? Well neither did my favorite computer-store owner, and neither did two of my favorite software publishers, and neither did I, until **ST Applications** interviewed Mark White, head of Technical Support, at the Supra Corporation, probably the largest supplier of hard drives for the Atari ST.

It's all got to do with the SCSI (Small Computer Standard Interface) system that the ST and the Mac use, as compared to the slow freight system that IBM uses. Next month Mark will expand on that difference, and he'll have some details on Supra's new 10-megabyte drive with interchangeable disks. And, if Supra ships as promised, you'll get a hands-on description of that drive, right here.

Speaking of hands being on, we're in line for one of MegaByte Computers's Accelerator boards. In this case, the hands that will be on shall be those of Bob McDonald of McDonald's Computer Center in Orlando, Florida, because that board requires quite a bit of soldering. We'll have photos for you of what's involved.

Additionally, we'll be -- well, just stay with us. The future for ST users is going to be pretty exciting, no matter what the complainers on CompuWhine may tell you!

END

More Freeware from page 32

Before I close this month's column, I should express some disappointment over promising to be able to cover the various music players that have sprouted on GEnie and the other networks. Most of them require MIDI instruments for output, so I will not at this time be able to review them or the various "artists" who are putting up music files. I am looking into spending a day or two at my favorite Atari dealer (COZ Computers in Chicago) listening to his Casio CZ101 to provide a report on this burgeoning PD field. It'll be a nice change of pace from waiting for the laser prints! I also still hope to hear from the author of ARC.TTP before covering it.

Finally, I'd like to thank all of the authors who have responded to my letters, almost all of whom called me at their expense to answer some questions and just plain talk about Atari and its computers. These people lead me to believe that no computer will completely die just due to lack of sales. With any Atari, you'll never be an orphan as long as the shareware network exists and as long as their community supports them. Talking to any one of these programmers, I received that same good feeling you get at a users group meeting: **We are not alone.**

END

The Joys of Modula 2

by Sol Guber

This months column will explain how to put a short little dialog box on the screen and get information from the person who operates the box. The dialog box will be moderately adjustable and will work in all resolutions, as well as allow for different colors to be used in the borders. The text in the box is completely adjustable. The procedure will function much like the one that is used to generate an alert box, but this dialog box will return more information then just which button was pushed. It will also return about 25 characters of text. Best of all, it will be generated without using a resource development program, it will generate the proper data structure inside the program.

One of the major differences between high and low level languages is that portions of programs need to be written only once in higher level languages. The major difference between Modula and C or Pascal, is the amount of inspection that is performed to be sure that the modules are used properly. There is a comparison of the information that is being transferred from the calling program to the library program. To use this month's program as an example, there are two sections to it, besides the little 15 line program that calls the procedure **Infobox.** The two portions are the **DEFINITION** Module and the **IMPLEMENTATION** Module. If you look at **Figure 1,** you can see that there is very little information about the function in the definition module and is much easier to write than the major program. The definition module states that there are three variables that will be used in the program, and there is a procedure that is exportable from it. The process of being exportable is the same as defining something **EXTERNAL** in C. Other programs can use this function and the variables that are initiated in it. The listing of several of the variables used in the program was also done in the **DEFINITION** module to make them external. They are then treated differently in memory, and are always accessible and will not be destroyed when the function is finished. The listing was also done to show how they were constructed and to allow for modifications in other programs.

The generation of both definition and implementation modules denotes that there is a very strong and well defined interface that is needed between the calling program and the library module. However, the only thing that a programmer needs to know to use the function Lineout is that a string being sent to it is a parameter. Once the Definition module is compiled, it turns into a .SYM file, with a special number written in it. When the implementation file is compiled, that number is also written into the .LNK file as part of the header.

When a program that uses the module is compiled, the first action of the compiler is to check that the function

is properly used. Both the number and the type of parameters must be correct. If the program compiles correctly, there is a number written into the .LNK file that comes from the .SYM file. When all of the sections for the program are linked, the last step before the program is run, there is a check that the number from the .SYM file corresponds to the .LNK file. If this is true, then the program is linked. If it is not, then there is an error message generated.

There is a bonus to this complexity. Once a .DEF file is written and compiled, it can be considered the standard. If the implementation module follows the rules set up in the .DEF file, then it can be modified and changed many times without affecting the original program. The only thing that the external world needs to know is that there is a string that is being returned from the function Lineout. For example, our dialog box has two buttons and one line of text that is not changeable, and one line of editable text. It is of a certain size, and the colors are selectable. If I decided to make three buttons and three lines of text, only one that would be editable, the function would still follow the rules and could be compiled with the same .DEF number. Thus once the .DEF file is fixed, the user does not need to know any of the details of how the program works.

Essentially what MODULA-2 allows you to do is make extensions to the language and to treat them as parts of the langauge. There is a function in the AESForms library to make an alert box. The details needed to pass the proper information that is given, but there is no way of seeing how the box is constructed. However, if the information is properly sent, then the function works. If you wish to make your own type of dialog box, and make your own library, then this is entirely possible.

The object which I call an **Infobox** can be created more easily using the Resource development kit that comes with the developer's version of Modula-2, but there is little documentation available on how to get information from dialog boxes and what they really do. A dialog box is a series of linked objects, and the linkage is shown in **Figure 1.** The actual structure of an object has been discussed in an earlier column, so what I will be doing in this one is just calling the object RECORD from GEMAESbase. I will also be calling the TEDInfo Record from the same file, without showing the structure in the program. For our purposes, it is not needed and can be a hidden part of the program.

The structure of the Object has many pieces. The first three numbers show the relationship between the object, and its children. The rest of the information explains what type of object it is, and how it is to be drawn upon the screen. Besides information on what kind of object it is, there is also data about the various things that can be done with the object (like it being an Exit or Default or Editable object), and the state of the object (Normal, selected, checked, etc.). There is also information about the position of the object. To add to the complexity, by definition, the children are always within the parent, so that every child's position must be given relative to the parent's. This can be easily seen by examining the numbers in the procedure Choice and the relationships shown there.

Another sign of the complexity of the object that is used for the dialog box, is the meaning of the .SPEC section of the object. It can have two meanings, one being the address for another object, and the other being 'packed' information about the object. The address information is used for the TEDINFO's and for the strings. When it is used as information there are five sections to the .SPEC portion of the object. The five parts are divided as follows with the dashes standing for the number of bits that are used:

```
+----+----+-+---+----+
| a| b | c| d  | e |
+----+----+-+---+----+
```

The high four bits ("a") are used for the border color with the values ranging from, 0 to 15. The next four bits ("b") are for the text color with the same range of 0 to 15. The 'c' bit indicates how the text is to be written with 0 being transparent and 1 being the replace mode. The next three bits ("d") indicates the fill pattern that is to be used with 0 being a hollow fill (no fill) and 7 being a solid fill with the values of 1-6 being patterns of increasing dark-ness. The last four bits ("e") are the objects inside colors, with values ranging from 0 to 15.

Besides the dialog box data, there is the TedInfo RECORD which is also intricate. The first three portions of the record contain the address for the various types of text that is needed for an Editable object. There is the address of the text itself, the template, and the valid string. The are certain codes that can be used in the valid string that allow only distinctive letters to be typed. These codes are shown in **Figure 4.** You can define the font to be used, the justification, as well as the length of the string and template. For our example, a "X" is used in the template so that any character can be typed.

The programming to generate our dialog box is straight forward. There are several procedures that are called to set up the initial values of the object. The object itself has six parts, a graphic box, an invisible box, two buttons, a line of text, and an editable line. The various flags and states were set up to correspond to normal dialog boxes.

Once the standard box was set up, the procedure CHOICE is called to individualize the dialog box. It is then modified for the resolution that is to be used. When a Infobox is made up using the Resource kit, the values are not put into the resource (.RSC), before it is called into the system. Once it is called, a check made to see what resolution the monitor is in, and the values in the coordinates are modified to correspond to the resolution. Also, the placement on the screen for the dialog box is not selectable. It is fixed when the dialog

box is made. I thought that this would be a nice little option. Once these choices are made, the values for the proper positioning is calculated and then put into the object.

The next part of the program is called Stringit. It is here that the various texts are put into the objects. The buttons expect a eight character text, but there is no checking to see it is done correctly. The top line expects a certain line of text and then the TEDInfo is set up. Once all of this has been done, the object is ready to be used.

LineOut is the procedure that does the actual drawing. It put the object where it is specified from the head of the object (box[0]) and then draws the object using the address of box[0]. It waits for a response, which the user pressing one of the two buttons. (This information which button was pressed, is not returned to the programmer). The string that was typed for the TEDInfo is found in info[3], the first part of the TEDInfo and it is returned to the programmer to use as he wishes.

The actual program itself just calls Lineout and a dialog box is

put on the screen with the sentence that was typed in. The system waits until a key is pressed and then returns everything. This function can now be used in any other program and is a convenient, quick method of retrieving information using the GEM interface.

Figure 1

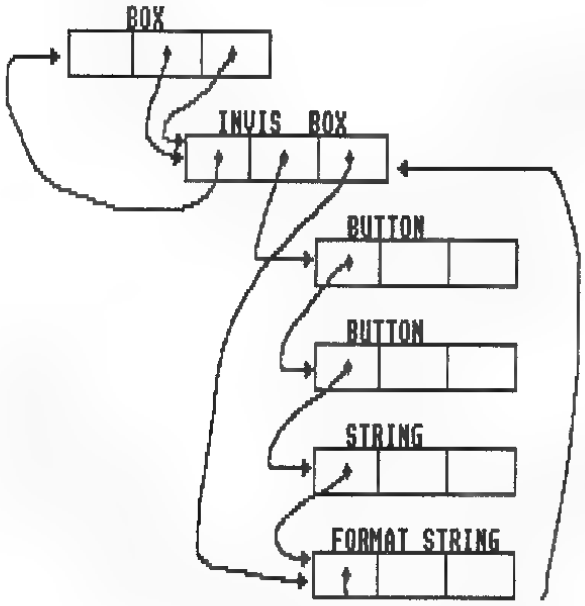


Figure 2

```
DEFINITION MODULE Infobox;

IMPORT GEMAESbase;

VAR
   box:ARRAY[0..5] OF GEMAESbase.Object;
   ted:GEMAESbase.TEdInfo;
   info:ARRAY[0..5],[0..25] OF CHAR;

PROCEDURE Lineout(VAR a:ARRAY OF CHAR);

END Infobox.
```

Figure 3

```
IMPLEMENTATION MODULE Infobox;

IMPORT GEMAESbase;
FROM GEMAESbase IMPORT GraphicBox,None,GraphicInvisibleBox,
     GraphicButton,Selectable,Exit,GraphicString,GraphicFormattedText
     Outlined,Normal;
FROM TextIO IMPORT WriteString,WriteLn,ReadString,ReadInt ReadHex;
FROM AESForms IMPORT FormDo,FormDialogue;
FROM AESResources IMPORT ResourceSetAddr;
FROM AESGraphics IMPORT GrafHandle
FROM AESObjects IMPORT ObjectDraw;
FROM SYSTEM IMPORT ADR,ADDRESS;

VAR
   i,j,k:INTEGER;
   a:ARRAY[0..25] OF CHAR,
   ch:CHAR;
   xrez,yrez,xq yq:INTEGER;

PROCEDURE Init;
BEGIN
   box[0].next:=-1;box[0].head:=1;box[0].tail:=1;
   box[0].type:=GraphicBox;box[0] flags:=None;box[0].state:=Outlined;
   box[1].next:=0;box[1].head:=2;box[1].tail:=5;
   box[1].type:=GraphicInvisibleBox;box[1].flags:=None;
   box[1].state:=Outlined;
   box[2].next:=3;box[2].head:= 1;box[2].tail:=-1,
   box[2].type:=GraphicButton, box[2].flags:=Selectable+Exit;
   box[2].state:=Normal;
   box[3].next:=4;box[3].head:=-1;box[3].tail:=-1,
   box[3].type:=GraphicButton; box[3].flags:=Selectable+Exit;
   box[3].state:=Normal;
   box[4].next:=5;box[4].head:=-1;box[4].tail:=-1;
   box[4].type:=GraphicString; box[4].flags:=None;box[4].state:=Normal;
   box[5].next:=1;box[5].head:=-1;box[5].tail:=-1;
   box[5].type:=GraphicFormattedText; box[5].flags =28H;
   box[5].state:=Normal;
END Init;

PROCEDURE Choice;
BEGIN
   WriteString("What resolution will be used");WriteLn;
   WriteString(" 1 - LOW ");WriteLn;
   WriteString(" 2 - MED ");WriteLn;
   WriteString(" 3   HIGH ");WriteLn,
   ReadInt(i);WriteLn;
   IF i=3 THEN xrez:=8;yrez:=16;
      ELSIF i=2 THEN xrez:=8;yrez:=8;
      ELSE xrez:=4;yrez:=8;
   END;
   WriteString("Which quadrant of the screen    1  2  ") WriteLn,
   WriteString("                                4  3  ");WriteLn;
   ReadInt(i);WriteLn;
   IF i=4 THEN xq:=5;yq:=10*yrez,
      ELSIF i=3 THEN xq:=40*xrez+5;yq:=10*yrez;
      ELSIF i=2 THEN xq:=40*xrez+5,yq =0;
      ELSE xq:=5;yq:=0;
   END;
   box[0].x:=xq,box[0].y:=yq+2;box[0].width:=35*xrez+box[0].x;
   box[0].height:=11*yrez,
   box[1].x:=16;box[1].y:=8, box[1].width:=33*xrez;box[1].height:=78;
   box[2].x:=48;box[2].y:=60; box[2].width:=10*xrez;  box[2].height:=12;
   box[3].x:=150;box[3].y:=60,box[3] width =10*xrez,  box[3].height:=12;
   box[4].x:=24;box[4].y:=25, box[4] width:=24*xrez;  box[4].height:=8;
   box[5].x:=24;box[5].y:=37; box[5].width.=24*xrez;  box[5] height =8,
END Choice;

PROCEDURE Stringit;
VAR i1:CARDINAL;
BEGIN
   WriteString("What is the color for the box 0 15');WriteLn,
   ReadHex(i1);WriteLn;box[0].spec:=ADDRESS(LONGCARD(i1)+3100H);
   box[1] spec:=12100H;box[2].spec:=ADR(info[0]);
   box[3].spec:=ADR(info[1]);box[4].spec =ADR(info[2]);
   box[5].spec:=ADR(ted);
   WriteString("What is the name of the left button? "),
   ReadString(info[0]);WriteLn;
   WriteString("What is the name of the right button? ");
   ReadString(info[1]);WriteLn;
   WriteString("What is the top line? ");WriteLn;
   ReadString(info[2]);WriteLn;
   ted.ptext:=ADR(info[3]);ted.ptmplt:=ADR(info[4]);
   ted.pvalid:=ADR(info[5]);ted.font:=3;ted.resvd1:=6;ted just:=0;
   ted.colour:=2183H;ted.resvd2:=0,ted.thickness:=255,
   ted.txtlen:=25;ted.tmplen:=25;
   info[3]:="                         ";
   info[4]:="_________ _________  ";
   info[5]:="XXXXXXXXXXXXXXXXXXXXXXX";
END Stringit;

PROCEDURE Linecut(VAR b:ARRAY OF CHAR);
BEGIN
   Init;Choice;Stringit;
   WITH box[0] DO
     FormDialogue(1,0,0,0,0,x,y,width,height);
     FormDialogue(2,0,0,0,0,x,y,width,height);
     ObjectDraw(ADR(box[0]),0,5,0,0,640,320);
     i:=FormDo(ADR(box[0]),0);
     FormDialogue(3,0,0,0,0,x,y,width,height);
     FormDialogue(4,0,0,0,0,x,y,width,height);
   END;
   FOR i:=0 TO 25 DO
     b[i]:=info[3,i];
   END;
END Linecut;

END Infobox.
```

Figure 4

```
MODULE Part23;

FROM Infobox IMPORT Linecut;
FROM TextIO IMPORT WriteString,WriteLn,Read;
FROM AESApplications IMPORT ApplInitialise,ApplExit;
FROM VDIControls IMPORT ClearWorkstation;

VAR
   a:ARRAY[0..25] OF CHAR;
   ch:CHAR;
   i:INTEGER;

BEGIN
   i := ApplInitialise();
   Linecut(a);WriteString(a);WriteLn;
   Read(ch);
END Part23.
```

Figure 5

```
                Template letters for TEDInfo structures
9 - allow only digits? 0 - 9
A - allow only uppercase A - Z plus space
a - allow upper- and lowercase A - Z plus space
N - allow 0   9 and uppercase A - Z plus space
n - allow 0 - 9 and upper- and lowercase A - Z plus space
F - allow all valid DOS filename characters, plus ? * :
P - allow all valid DOS filename characters, plus \ A * :
p - allow all valid DOS filename characters, plus \ :
X - allow anything
```

Prologomenon

by Joseph Schmuller

In the last installment, we developed doctor", a program which gives the appearance of processing everyday language input from the keyboard and responding intelligently. Of course, it does no such thing. What it really does is 1) takes a line of input, 2) turns this line into a list of words, 3) tests it to see if any of the words are a member of one of several categories, and then 4) responds on the basis of the outcomes of the tests. There is no linguistic analysis involved, as "doctor" has no sense of grammar.

Grammars

This month, we'll examine one way in which Prolog can be used to do grammatical analysis on a set of words. What is "grammatical analysis"? Imagine a machine which takes a sequence of symbols as its input, examines the sequence, and outputs the response "yes" if it judges the sequence to be "valid", or "no" if it judges the sequence to be "invalid". Inside this machine is a set of rules which actually make the judgment. This set of rules is the "grammar", and the process that the rules follow is "grammatical analysis". Notice that we haven't said anything yet about "language"; all we've mentioned so far are judgments on sequences of symbols. You can construct a grammar which specifies that the letters "a" and "b" are the only two symbols allowed into the machine and that within any valid sequence, these symbols must alternate. In that case, "abab" and "baba" would be valid sequences, while "bbba" and "abba" would not.

Each time we submit a computer program to a compiler, we are sending a sequence of symbols into a set of rules. The rules then decide if the sequence is valid or not. (I.E., if we've written anything which is contrary to the rules of the particular language we happen to be programming in).

Analogous to the rules for computer languages, the languages which humans use to communicate with one another also have rules which delineate what is valid from what is invalid. These rules were not set down in advance, but instead evolved as the languages did. One task in the field of linguistics is to listen to people as they communicate and then try to infer what these rules of "correctness" must be. We'll begin our study with simple rules for constructing valid sequences in English that we commonly call "sentences".

I think we would all agree that this is a sentence:

The boy chases the dog.

and this is not:

The boy the dog.

The difference between these two sequences (let's call them "strings") is that the second one has no verb. "Verbs","nouns",etc., are all "parts of speech", and we can replace each word in the first string with the "part of speech" which it happens to be:

determiner noun verb determiner noun.

I hope this stuff about "parts of speech" doesn't bring back unpleasant memories of grade school (isn't that also called "grammar" school? Hmmm ...). Anyway, a determiner followed immediately by a noun is a combination called a "noun-phrase". So if we substitute this larger combination into the string of parts of speech, we'll have:

noun-phrase verb noun-phrase.

Now, a verb followed immediately by a noun-phrase is a combination called a "verb-phrase", so that our string could now be written as:

noun-phrase verb-phrase.

A noun-phrase followed by a verb-phrase constitutes a "sentence". There are many different types of valid sentences in English, and this is one of them. Thus, we've shown that the word-string "The boy chases the dog" is a sentence by starting with the words of the string, replacing them by the appropriate parts of speech, combining the parts of speech into larger units, and showing that these larger units, when taken together, make a sentence.

We can also move in the other direction. That is, we can show that the generic unit "sentence" can be broken down into progressively smaller units, terminating with individual words. We can represent the rule that a sentence can be broken down into a noun-phrase and a verb-phrase with this notation:

```
sentence --> noun_phrase,verb_phrase.
```

Instead of saying "broken down into" let's say "written as". The preceding rule shows that a sentence can be written as a noun-phrase and a verb-phrase. To be more complete, the rule should be:

```
sentence --> noun_phrases,verb_phrase,terminator.
```

We use "terminator" to represent the fact that a sentence must end with a period. In general, we'll refer to rules like this as "rewrite rules". The rewrite rule for a noun-phrase is:

```
noun_phrase --> determiner, noun.
```

This rule shows that a noun-phrase can be written as a determiner followed by a noun. The rewrite rule for a verb-phrase is:

```
verb_phrase --> verb, noun_phrase.
```

Our rewrite rules can go down one more level:

```
determiner --> [the].
noun --> [boy].
noun --> [dog].
verb --> [chases].
terminator --> ['.'].
```

Note the brackets around the words (and around the period in the last rule). The bracketing distinguishes them from parts of speech (and from larger units). These are called 'terminals'; they cannot be broken down further. Units like nouns, verbs, determiners, noun-phrases, and verb-phrases are called "non-terminals", as they can be broken down further.

Definite Clause Grammars

The above set of rewrite rules is called a grammar. The notation we've been using is called a "definite clause grammar" or DCG. It's a grammar because it shows how words in a particular vocabulary can combine to form one kind of correct sentence. Actually, it's more accurate to say that the set of rules we've written (i. e., the grammar) can be used to test whether or not a word string constitutes one kind of sentence. This testing is called "parsing". Of course, the words in the word string have to be among the terminals in the grammar. The string:

the girl chases the dog.

would not be classified as a sentence because the grammar (as it stands) doesn't recognize "girl". The string:

the boy likes the dog.

would also not be classified as a sentence because the grammar doesn't recognize "likes". If we add:

```
noun --> [girl].
verb --> [likes].
```

we'll considerably expand the number of word-strings that our DCG will classify as sentences.

DCGs in Prolog

Prolog turns out to be a very easy language in which to implement functions for constructing DCGs. In fact, XPRO has already done this for us. Take a look at the XPRO User's Guide. Page 19 shows that "-->" is an xfx operator whose precedence is 1200. If we type all of the aforementioned rewrite rules into a file, and then con-

sult this file, we'll be able to use the built-in predicate "phrase" to see if a list of words is a sentence or not:

```
?-phrase(sentence,[the,boy,chases,the,dog,.]).
 Yes.

?-phrase(sentence,[the,boy,the,dog,.]).
 No.
```

"phrase" takes two arguments, a non-terminal and a list. "phrase" succeeds if the list is an example of the non-terminal:

```
?-phrase(noun_phrase,[the,boys]).
 Yes.

?-phrase(verb_phrase,[the,likes]).
 No.
```

"phrase" also works like this:

```
?-phrase(sentence,X).
```

If you hit the semi-colon key after each instantiation, "phrase" will instantiate X to every possible sentence in the DCG.

We can make things easier for ourselves by writing a function that will allow us to just enter the input string and have our grammar parse it. Here's one way to write it:

```
test :-
      nl,write('Enter the string'),
      nl,write('>'),
      readl(STRING),
      phrase(sentence,STRING),
      nl,write('This is a sentence').

test :-
 nl,write('This is not a sentence').
```

Last month, we said that the predicate "readl" is handy for natural language processing; the above function shows why.

Extensions to DCGs

Suppose we add the following rewrite rules to the DCG we've been using:

```
noun --> [boys].
verb --> [chase].
```

These strings will be identified as sentences:

```
the boy chases the dog.
the boys chase the dog.
```

But so, unfortunately, will these:

```
the boy chase the dog.
the boys chases the dog.
```

These last two examples are wrong because the nouns don't agree with the verbs in terms of number. The first has a singular noun ("boy") with a plural verb ("chase"), while the second has a plural noun ("boys") with a singular verb ("chases"). How can we set up our grammar to make the nouns and verbs agree?

One way would be to have one set of rules for singular sentences and another set for plural sentences. As an exercise, you might try setting up such a grammar. If you do, you'll see that it will have to contain quite a few rules. We can cut down on the number of rules we need if we use arguments to denote numerical agreement. In other words, our rules can look like this:

```
sentence-->noun phrase(NUMBER),verb phrase
(NUMBER),terminator.
noun phrase(NUMBER)-->determiner(NUMBER),
noun(NUMBER).
verb_phrase(NUMBER)-->verb(NUMBER),noun_
phrase(_).
noun(SINGULAR) --> [boy].
noun(PLURAL) --> [boys].
noun(SINGULAR) --> [dog].
verb(SINGULAR) --> [chases].
verb(PLURAL) --> [chase].
determiner(_) --> [the].
terminator --> ['.'].
```

The anonymous variable on the right side of the verb-phrase rewrite rule allows the grammar to classify both of these as sentences:

```
the boy chases the boy.
the boy chases the boys.
```

The anonymous variable in the argument of the determiner's rewrite rule lets us use "the" with both singular and plural nouns. By the way, once you use an argument with a term within a DCG, that term must always have that same arity within that DCG. (Remember that arity means "number of arguments").

Procedural Stuff

So far, everything we've said about DCGs has been in the declarative realm. We can extend the range of things that DCGs work with by adding some procedural mechanisms to our rewrite rules. A procedure is added to a rewrite rule by placing it inside curly brackets and adding it to the right-hand side.

Here's a classical example. Suppose we want to refer to a calendar date such as October 23rd or August 10th. We know that a day of the month cannot be less than 1 or greater than 31, and also all the acceptable names for months (let's limit them to three-letter abbreviations).

We can set up the following rewrite rules for date-phrases:

```
date--> month(MONTH),[NUMBER],{integer(NUM-
BER),NUMBER >=1,NUMBER <=31}.

month(M)--> [M],{member(M,[jan,feb,mar,apr,may,
jun,jul,aug,sep,oct,nov,dec])}.
```

The statements inside the curly brackets in the first rule tell us that the date-number has to be an integer within the specified limits. The second rule declares that the argument to "month" will be the terminal. Generally, this is a pretty nifty little trick within DCGs: the use of a terminal as an argument. The "member" function in the curly brackets specifies what the argument to "month" has to be in order to be an acceptable month-name. In order for this to work, of course, "member" has to be in Prolog's database.

This example was adapted from Pereira and Warren's landmark paper on DCGs, which appeared in the journal "Artificial Intelligence", volume 13 (1980), pages 231-278. You can find this journal in most university libraries. The papers in it are, as a rule, highly technical, but Periera and Warren's paper is surprisingly readable.

We can expand on Pereira and Warren's example by putting more structure into our rules for calendar dates. Our rules right now will classify "feb 30" and "nov 31" as legitimate. Since we know that these dates don't exist, we have to add something. Let's change the "[NUMBER]" terminal in the "date" rule into a functor called "numeric", which takes two arguments "NUMBER" and "MONTH". Now, the "date" rule looks like this:

```
date--> month(MONTH),numeric(NUMBER,
MONTH),{integer(NUMBER),NUMBER >=1,NUM-
BER =<31}.
```

We'll need some rewrite rules for "numeric":

```
numeric(N,M)--> [N],{N=>30,member(M,[sep,apr,
jun,nov])}.
numeric(N,M)--> [N],{N=28,M=feb}.
numeric(N,M)--> [N],{N=31,member(M,[jan,mar,
may, jul, aug, oct, dec])}.
```

This grammar will do the trick.

There's another way to expand this date-grammar. We often abbreviate "first" as "1st" and twenty-third as "23rd". We don't, however, say "1th","2th" or "11st". Let's add the numerical suffixes "st", "nd", "rd", and "th" into our date-phrases, making sure that the right suffix goes with the right number. We'll change the "date" rule once more, so that it looks like this:

```
date--> month(MONTH),numeric(NUMBER,
MONTH),suffix(NUMBER),{integer(NUMBER),
NUMBER >=1,NUMBER <=31}.
```

and add these rewrite rules:

```
suffix(N)Z--> Z[].
suffix(N)Z--> Z[st],{N=1;N=21;N=31}.
suffix(N)Z--> Z[nd],{N=2;N=22}.
suffix(N)Z--> Z[rd],{N=3;N=23}.
suffix(N)Z--> Z[th],{(4=<N,N=<20);(24=<N,
N=<30)}
```

The first "suffix" rule lets us continue to write dates as before, with no suffix. (Note the use of the semi-colon within the curly brackets of each rule.) Each semi-colon should be read as "or". In the last rule, we use parentheses along with the semi-colons in an intuitive way. They help us denote that in order to use the suffix "th", N has to be between 4 and 20, or between 24 and 30. We haven't used this type of syntax yet, but it is a part of Prolog.

Here's an exercise for you. Our date-grammar will now classify "oct 23 rd" as a legal date only if there is a space between the "23" and the "rd". Write a function "test_date" that takes a date the way we'd usually write it ("oct 23rd"), without the space between the number and the suffix), and tests to see if its a legal date. This function should work like this:

```
?- test_date.
 > oct 23rd.
This is a legal date.

?- test_date.
 > nov 31st.
This is not a legal date.
```

Summing Up

Thus far, all we've done with DCGs is specify the sequences that qualify as valid sentences of a particular type, and the sequences that qualify as valid calendar-dates. We've also looked at a built-in predicate which tests sequences to see if they qualify in either case.

But we haven't really made these sequences do anything. What we've done up to this point is analogous to two people talking to one another. Each time one person says something, the other one decides only if the first person's utterance was grammatical or not. Neither one acts on what the other says.

How can we change this? That is, how can we type an English sentence into Prolog, and have Prolog do something as a result of the content of the sentence? Here's an example. Suppose we have built a Prolog function called "construct". The purpose of this function is to examine a sequence of words. If the sequence is a sentence, the function pulls out the subject, verb, and object, constructs a fact (with the verb as the functor and

the subject and object as arguments), and asserts that fact into Prolog's database. Ideally, "construct" should tell us the fact it has asserted, or in the case that the sequence is not a sentence, it should tell us that, too. Here's how it would be used:

```
?- construct.
 > the boy chases the dog.
Just asserted: chases(boy,dog).

?- chases(X,Y).
 X = boy Y = dog

?- construct.
 > the boy the dog.
This is not a sentence.
```

If we can get Prolog to do this, we can start to develop programs which understand at least a small subset of English. That is, we type an English sentence, and Prolog will represent the information in that sentence in a form which it can manipulate and use to answer questions (which might also be stated in English). Another thing we can do, once we've gotten Prolog to understand some English phrases, is express Prolog predicates as English sentences and have Prolog execute those predicates.

Just how do we get Prolog to pull the important features out of sentences and work with them further? I'll show you that next month. In the meantime, it would be an interesting exercise for you to try to figure it out before the next issue comes off the presses. One hint: we'll have to use DCGs in a slightly different way than we have up to this point. Another good exercise would be to set up DCGs for other types of sentences, such as questions. XPRO has a file called "converse" which does this (but without using arguments). One final exercise, and then I'm out of here: after you've consulted a DCG file, do a listing for each non-terminal. Examine the listing: doesn't something look wrong? Aren't there too many arguments? What does it all mean?

See you next month.

More Desktop Publishing from page 35

This program has several detrimental bugs in the current version. I have repeatedly bombed the program while attempting to size graphics in a document. **Word Up's** footnote option is buggy and I have had repeated system lockups for no particular reason. I talked to Mike Fulton at Neocept and he assures me they are working to fix current shortcomings in this program.

Drawbacks in this program include an inability to write text over graphics (however there is a "work around" for this), unuseable ASCII text output, and slow scrolling speeds as a result of the many graphics and text sizes used.

If you are patient it is possible to discover what currently bombs the program and work around it. I understand another upgrade will be available shortly and I look forward to a version that fixes some of these bugs. Out of the three Pseudo Publishing programs I think **Word Up** is the best value for your money.

For more information contact Neocept, Inc., 903 Camino dos Rios, Thousand Oaks, CA 91630. To order call: (800)666-8766. Word Up retails for $79.95.

Final Notes

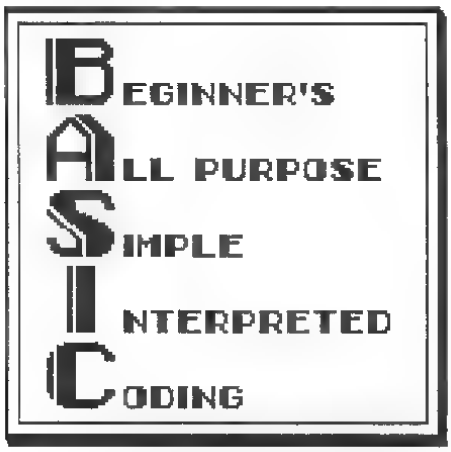
So there you have it - 3 Pseudo Publishing programs and an idea of what they do. Before closing this month I wanted to acknowledge several individuals this article would not have been possible without.

Special thanks go to Mike Zielinski, the owner of Software 1st, in Santa Rosa, California. Mike made copies of **Microsoft Write** and **1st Word Plus** available to me free of charge to write this review. The repeated help of Mike and his staff have given me insight and perspective in managing this column. Thanks Mike!

If it was not for reviewers copies from companies like Neocept, Inc. much of what you read in any magazine would never get written. Over two months of testing and using **Word Up**, Shelby Moore III and Mike Fulton have been instrumental in answering questions and providing upgrades. I believe it must be particularly traumatic for a programmer to give free reviewers copies of his software to magazines and then submit to a dissection of their product. Thanks guys!

Publishing Tips

Not a single idea for special uses of DTP software winged my way this month. For the present I'll be optimistic and hope to hear from you during July. C'mon gang! Share what you have learned with other ST owners. See Ya Next Month!



Gumdrops, Rolling Stones, and Moss.

By Eric Thornton

This program was written in ST Basic. The old one, not the new one. (Are they shipping the new one with the machines yet?) The program generates a simple graphic of a falling ball, with shading and lighting effects. You might have to duck though, because the ball is falling towards you.

Variable Usage:
C
S
D
XD
CX
CY

The output window is opened, set to full size, and cleared by the command in line 1. The subroutine at line 1000 is called. It sets the fill perimeter to off (see below).

The size of the ball starts at zero and moves up by one every time until it reaches 120. CX and CY are assigned values which position the center of the ball on the screen. The subroutine which draws the ball is called. Next S repeats the size loop increasing the size of the ball.

I saved some time by having the newest biggest ball drawn over the top of the last, smaller ball. Thereby not having to erase the previously drawn ball from the screen. After the S loop completes (reaches maximum) the program ends. You can stop the program execution before the program ends by pressing the CONTROL-C key combination.

Lines 10 thru 50 is the old slow way to do the same job as the subroutine BALL. It's still here so you can try running it and compare its speed to the final version. I won't bother to describe the old routine, just the new one (BALL).

Line 1000: uses a VDI call to set the fill edge to off. This means that when you use the pcircle command the filled circle drawn has a solid line around it. Well after the call at line 1010, there will not be that solid line around the edge of filled circles. It messed up the appearance of the ball which uses overlapping filled circles of different shades of grey to similate light hitting the ball.

To call GEM VDI the program must pass arguments in order starting at the memory location pointed to by the built in variable CONTRL. 104 is the opcode for V SFperimeter.

Line 2000: starts the subroutine (called ball) which draws the ball given the variables for location and size defined earlier.

D is a loop which runs backwards, that is, it starts with a value of eight and decreases each time. The Next D

command is executed and stops when the value of D is less than 1. The backwards stepping is caused by the negative step value after the For... To... part.

The fill index is set from D and the fill style is dot pattern (2). Each fill grey is drawn at a calculated position which is a little further up and to the right of the previous one. They could have been concentric circles, except it would look like you were standing in the middle of the light source.

All in all a simple little useless program, right? Wrong, it demonstrates that sometimes re-writting something can improve the look and speed, and that it's not chiseled in stone just because it works. Until next time, take care.

Balldrop bas.

```
1     openw 2:fullw 2:clearw 2
2     gosub 1000
3     c=8
5     for s=0 to 120:cx=400:cy=65: gosub ball:next s:end
10    for xd=80 to 0 step -10
20    color 1, 1 ,1,c,2:c=c-1
30    pcircle 400-(xd/4),10+(xd/4),xd
40    next xd
50    end
1000  c=contrl:poke c,104:poke c+2,0:poke c+6,1
1010  poke intin,0:vdisys(0):return
2000  ball:for d=8 to 1 step -1
2010  color 1,1,1,d,2:pcircle cx-((d*s)/4) ,cy+((d*s)/4),d*s
2020  next d:return
```

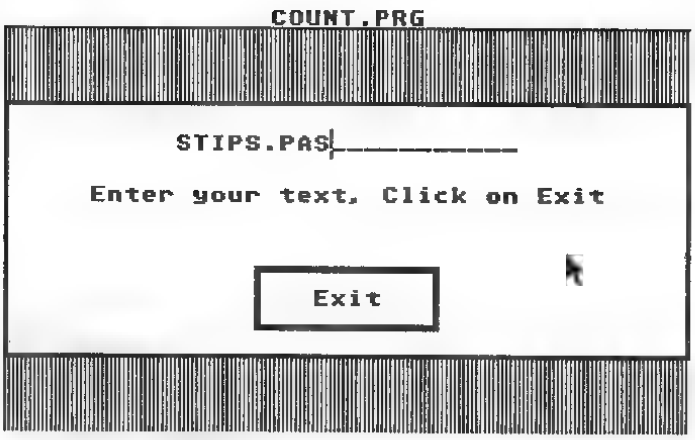
END

More Picshell from page 8

```
      Else
        L$(I%)=""
      Endif
      Mid$(X$,13,1)=L$(I%)
      P$="*.PRG"
      P$="*.TOS"
      If Instr(X$,P$)=0 And Instr(X$,P1$)=0 And L$(I%)<>"\" Then
        File$(I_%)=""
      Else
        File$(I_%)=X$
        Inc I_%
      Endif
    Wend
    File$(I_%)=""
    Swap *Str.arr%,File$()
    *Num.%=I_%-1
  Return
  Procedure Quicksort(Str.arr%,L%,R%)
    Local X$
    Swap *Str.arr%,A$()
    @Quick(L%,R%)
    Swap *Str.arr%,A$()
  Return
  Procedure Quick(L%,R%)
    Local Ll%,Rr%
    Ll%=L%
    Rr%=R%
    X$=A$((L%+R%)/2)
    Repeat
      While A$(Ll%)<X$
        Inc Ll%
      Wend
      While A$(Rr%)>X$
        Dec Rr%
      Wend
      If Ll%<=Rr%
        Swap A$(Ll%),A$(Rr%)
        Inc Ll%
        Dec Rr%
      Endif
    Until Ll%>Rr%
    If L%<Rr%
      @Quick(L%,Rr%)
    Endif
    If Ll%<R%
      L%=Ll%
      @Quick(L%,R%)
    Endif
  Return
  Procedure Original
    Cls
    For J=0 To 15
      C=Xbios(7,J,Sav%(J))
    Next J
    C=Xbios(5,L%-1,L%-1,W:Res)
  Return
  Procedure Newpal
    For J=0 To 15
      C=Xbios(7,J,Pal2%(J))
    Next J
  Return
  Procedure Degas
    Discard=34
    Z$=Input$(2,#1)
  Return
  Procedure Neatpal
    Void Xbios(7,0,0)
    Void Xbios(7,3,&H770)
    Void Xbios(7,15,&H770)
  Return
```

END

STips.Pas



by Chuck Adams

Compulogic Development Corp.

Last month, I introduced a very simple program. This program, called **Count**, uses an input Dialog box to enter a string. It then counts the total number of characters, the number of commas, spaces, and periods.

The counts are displayed using an output Dialog box. This month, I will concentrate on the Gem calls which create Alert boxes and dialog boxes. My discussion will be limited to an introduction to these calls, intending to provide a working knowledge of Alert boxes and Dialog boxes. Again, as last month, I will be using Personal Pascal.

As you will see when you run the program, I have added an Alert box.

This box prompts you to continue or to exit. I have further modified the program by "modularizing" it. The program is now divided into procedures. The main procedure calls three procedures: Input_Dialog,Output Dialog, and Exit_Alert. The Output_Dialog procedure calls another procedure Count Char. Count_Char evaluates the string. By dividing a program into procedures, you can (1) more easily reuse sections of the code in other programs, (2) divide the program into sections allowing more than one person to write code, and (3) createmore complicated and powerful programs.

The first difference you will notice when writing a program to use GEM, are the GEM calls Init_GEM and Exit_GEM. Init_GEM - is a function used to initialize GEM and allow you to use other GEM calls in your program. This function opens a virtual work station to the screen. This call requires no parameters.

Sample Call:

```
If Init_Gem=0 THEN BEGIN
```

If Init_GEM returns any value less than 0 then it failed. In the sample program, all of the procedure are called from within the IFTHEN loop. If this statement evaluates to false then the program ends.

Exit_GEM - is a procedure which will reset GEM allowing you to exit your program with "no unfinished business". This call will more then likely be your last operation. This call has no parameters.

Sample call:

```
Exit_GEM; {Closes virtual work space}
```

The Alert box is the least complicated and easiest GEM call to use. Alert boxes are typically used for simple decision making, pauses in the program, and error messages. An Alert box is executed by a call to a single function, Do_Alert. This function requires a

parameter made up of two components, a string and an integer. The string may have a maximum of 255 characters.

Sample call:

ExitVar := Do_Alert('[2][Enter another input string][Yes | No]',0);

ExitVar - The function Do_Alert returns an integer which is stored in this variable. The value of the integer indicates the button selected. The string is contained within the quotes.

The string is made of three parts. The first part [2] indicates that a question mark icon will be displayed in the Alert box (note 1). The second part [Enter another input string] is an instruction line. The last part of the string, [Yes | No] are the button names. Each button name is separated by a vertical bar.

The second component, the integer, designates the default button (if any). The integer may have the following choices:
0 - no default
1 - the first button is the default
2 - the second button is the default
3 - the third button is the default.

If the default integer exceeds 3 or the number of buttons used, unpredictable results may occur.

The Dialog box is the next most commonly used GEM call to communicate with the user. Dialog boxes are more complicated than Alert boxes, but provide many additional capabilities to the programmer. Unlike the Alert Box, a sequence of several steps is required to create and execute a Dialog box. I will discuss these steps in the order you will most likely use them.

New_Dialog - The first step is to create the Dialog box using the New Dialog call. This function reserves memory space for the Dialog Box descriptor. This function is of type Dialog_Ptr and is made up of 5 integer parameters. I will discuss each in turn using the sample program.

Sample call:

Input_Box := New_Dialog(7,0,0,40,15);

Input_Box - User defined as type Dialog_Ptr.
7 - The number of items to be included within the box. (PP recommends that this number always exceed the actual number of items to be included, because the components of the box itself count as items.)
0 - The horizontal position of the top left corner of the Dialog box in characters..
0 - The vertical position of the top left corner of the Dialog box in characters.
40 - The width of the Dialog box in characters.
15 - The height of the Dialog box in characters.

(For the sample program, I use the command Center_Dialog() later. This command automatically centers the Dialog box on the screen. Therefore any entries to the horizontal and vertical positions are superfluous.)

D_Color - This procedure is used to assist in computing the object color value for each of the items in Add Ditem. This call requires 5 parameters.

Sample Call:

Item_Color := D_Color(Black,Black,True,0,White);

Item_Color - Defined as type integer. This variable holds the value of the object color computed from the call to D_Color.
Black - The first parameter defines the color of the border. For the available colors, see note 3.
Black - The second parameter defines the color of the text. For the available colors, see note 3.
True - This parameter defines the text mode. True designates replace mode. The text will overwrite the background color. False would designate transparent mode. The text and background would be combined. (You should note the PP manual specifies the use of either a 0 or a 1 for false and true. You should use either true and false as indicated in the sample program or define a boolean variable.)
0 - The fourth parameter determines the pattern. 0 is no color, 7 is a solid color.
White - The last parameter is the fill color (see note 3).

Add_Ditem - After reserving space for the Dialog box, this call is used to add items to the box. This function requires 9 parameters and returns the index of the item within the GEM object structure.

Sample call:

Data_In := Add_Ditem(Input_Box,G_FText,Editable, 10,2,20,1,0,Item_Color);

Data_In - Defined as type integer. This variable will hold a unique identifier of the item within the dialog tree.
Input_Box - Pointer to the Dialog box.
G_FText - This parameter describes the object type (note 2). This object is a line of editable text or the input string.
Editable - An object flag (note 4) used to describe the characteristics of the item.

10 - The horizontal position of the top left corner of the item relative to the top left corner of the Dialog box, in characters.

2 - The vertical position of the top left corner of the item relative to the top left corner of the Dialog box, in characters.

20 - The width of the item in characters.

1 - The height of the item in characters.

0 - The width of the border around the item. 0 specifies no border. 1 to 127 thickens the border inward. -1 to -127 thickens the border outward.

Item_Color - Determines the item color. This value has been obtained from a previous call to D_Color.

Set_DText - is used to place text into the appropriate item within the dialog tree. This procedure requires 5 parameters.

Sample Call:

Set_Dtext(Input_Box,Instruct,'Enter your text, Click on Exit', System_Font,Te_Center);

Input_Box - Pointer to the Dialog box.

Instruct - Index of the item within the Dialog box. (This is the index obtained from the call to Add_Ditem.)

'Enter your text, Click on Exit' - The text to be entered to the item. This must be a character string no greater than 255 characters.

System_Font - This specifies the font size and type. The only choices for this parameter are System_Font and Small_Font.

Te_Center - This parameter justifies the text within the item borders. The only choices are Te_Left, Te_Right, and Te_Center. These descriptors will left justify, right justify, or center the text.

Set_DEdit - is a procedure used to describe an editable text field. This GEM call is made up of 7 parameters.

Sample Call:

Set_DEdit(Input_Box,Data_In,'_____
', 'xxxxxxxxxxxxxxxxxxxx',",System_Font,Te_Center);

Input_Box - Pointer to the Dialog box.

Data In - Index of the item within the Dialog box. (This is the index obtained from the previous call to Add_Ditem.)'_____' - This parameter, defined as a String[255], is the template for the editable field. In this case, the underbars reserve blank spaces for data entry. Normally characters entered in this string would appear on screen exactly as they appear here. The underbar is the only exception.

'xxxxxxxxxxxxxxxxxxxx' - This parameter, defined as a String[255], is the validation field for the editable field. The entry to this field will define the allowable input(note 5). In-valid entries will be ignored. In this case, x's will accept any printable character.

" - This parameter, defined as a String[255], is the initialization field. Any entry in this field will be displayed as a default when the Dialog box is displayed. The sample call has no default.

System_Font - This specifies the font size and type. The only choices for this parameter are System_Font and Small_Font.

Te_Center - This parameter justifies the text within the item borders.

The only choices are Te_Left, Te_Right, and Te_Center. These descriptors will left justify, right justify, or center the text.

Obj_SetState - is a procedure which defines the initial state of the item to be included in the Dialog box. This GEM call has 4 parameters.

Sample Call:

Obj SetState(Input Box,Exit But,Normal,False);

Input_Box - Pointer to the Dialog box.

Data_In - Index of the item within the Dialog box. (This is the index obtained from the previous call to Add Ditem.)Normal - This parameter defines the state of the item. Examples are Normal, Selected, or Disabled. These choices may be OR'ed together. The sample call is set to Normal.

False - The last parameter, a boolean, tells GEM if the item needs to be redrawn. This parameter will always be false the first time the Dialog box is displayed. This parameter should be defined True if you have already called Do Dialog and will make a call to Redo_Dialog.

Center_Dialog - is a procedure which will automatically center the Dialog box on the screen. This call many times will eliminate the need to calculate the position you want the Dialog box to appear.

Sample Call:

Center_Dialog(Input_Box);

Input_Box - Pointer to the Dialog box.

Do_Dialog - is the function which displays the dialog on the screen. This GEM call has 2 parameters.

Sample Call:

Choice : = Do_Dialog(Input_Box,Data_In);

Choice - This integer variable will hold the index of the item which returns control to the program from the Dialog box.

Input_Box - Pointer to the Dialog box.

Data_In - The index of the item the cursor should appear in. If there are no editable fields in the Dialog box, the index should be 0.

End_Dialog - This procedure will erase the Dialog box from the screen. Memory is still reserved for the dialog, it may be re-displayed by a call to Redo_Dialog. Only 1 parameter is required for this GEM call.

Sample Call:

```
End_Dialog(Input_Box);
```

Input_Box - Pointer to the Dialog box.

Get_Dedit - This is the GEM call to find the editable field within the Dialog box and transfer the contents to a String variable. This call uses 3 parameters.

Sample Call:

```
Get_Dedit(Input_Box,Data_In,DataStr);
```

Input_Box - Pointer to the Dialog box.

Data_In - Index of the item within the Dialog box. (This is the index obtained from the previous call to Add_Ditem.)

Data_String - This is a string variable defined String[255]. Get_Dedit will read the information in Data_In and copy the information to Data_String. The information contained in Data_String can now be interpreted and processed.

Delete_Dialog - This GEM call will release the memory space reserved for the Dialog box. This call only has one parameter. After executing Delete_Dialog, any subsequent use of this Dialog box can only be made by re-executing all of the calls used to create the Dialog box beginning at New_Dialog.

Sample Call:

```
Delete_Dialog(Input_Box);
```

Input_Box - Pointer to the Dialog box.

Additional explanation and detail for each of the above calls can be found in the Personal Pascal manual. Other commands you may want to learn are: Show_Dialog, Obj_State, Obj_Redraw, Obj_Flags, Obj_SetFlags, and Redo_Dialog.

An alternative to the above calls for the Alert box and the Dialog box is the use of a Resource Construction Set. PP provides the means to load the files created by the RSC. Atari has a Resource Construction Set available with the developer's package.

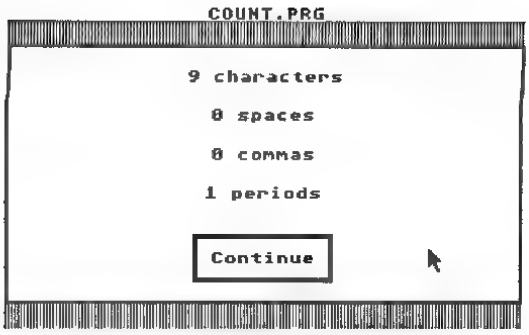
Note 1
Alert Icon Definitions
0 - No icon.
1 - An exclamation point in a diamond.
2 - A question mark in an inverted triangle.
3 - The word "STOP" in a stop sign.

Note 2
Object Type Definitions
G_IBox - The outline of a box.
G_Box - Same as G_IBox, but the box is filled with the fill color and pattern specified.
G_Text - A line of text specified by a subsequent call to Set_DText.
G_BoxText - Same as G_Text, except the text will have a border surrounding it.
G_FText - A line of editable text specified by a subsequent call to Set_DEdit.
G_FBoxText - Same as G_FText, except the text will have a border surrounding it.
G_String - A simplified version of G_Text. Border size, item width, and color are ignored. The text is always left justified.
G_Button - A simplified version of G_BoxText. The border width is set to 1 and the color parameter is ignored.

Note 3
Standard Color Values

0	White	9	L_White
1	Black	10	L_Black
2	Red	11	L_Red
3	Green	12	L_Green
4	Blue	13	L_Blue
5	Cyan	14	L_Cyan
7	Yellow	15	L_Yellow
8	Magenta	16	L_Magena

Note 4
Object Flags
None($00) - No flag set. This flag should be used when no flag is to be set.
Selectable($01) - Makes an item selectable.
Default($02) - Makes an item the default. This item will be selected by pressing the return key.
Exit_Btn($04) - This flag allows the item to exit the Dialog box and then return control to the program.
Editable($08) - This flag allows the item to be edited.
Radio_Btn($10) - This flag allows only one button to be selected. Other buttons with this flag will automatically be deselected.
Touch Exit($40) - similar to Exit-Btn except the command will be executed a soon as the mouse button has been pressed.



Note 5

Validation Characters for Set_DEdit

9 - Accept only decimal numeric digits.

A - Accept only upper case alphabetic characters, or a space.

a - Accept only lower case alphabetic characters, or a space.

N - Accept digits, upper case alphabetic character,s or a space.

n - Accept digits, upper and lower case alphabetic characters, or a space.

F - Accept any character that can be part of a GEM-DOS FILE name.

P - Accept any character that can be part of GEM-DOS PATH name.

p - Same as P, but do not accept any wild card characters.

x - Accept any printable character.

Miscellaneous News

You should be aware of the following bugs:

- When copying files to the same directory, a warning will be displayed of a name conflict. If you click on OK without changing the name, the operating system will copy the file over itself. The result will be a corrupt file.

- When using the file selector, the underbar character will cause the system to bomb.

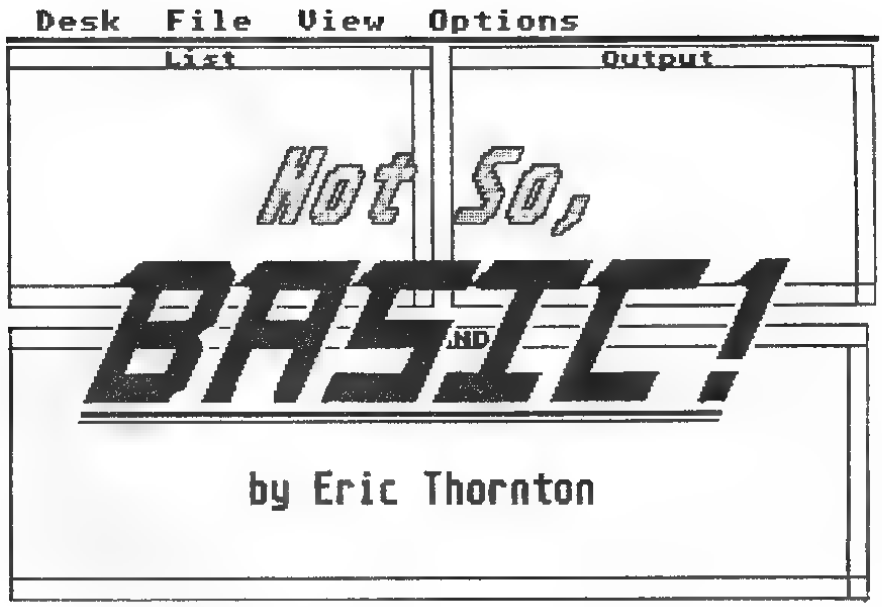
I recently called ICD for updates on Personal Pascal. A spokesman reported they have decided not to honor any of the free upgrade cards distributed by OSS. The latest version of PP is 2.02. To obtain this version as an upgrade, mail in your current copies of PP along with $10 to the following address: ICD, 1220 Rock St., Rockford, IL 61101. I also learned that version 2.01 is not compatible with the new ROMS. If you intend to install the new ROMS you may have to upgrade.

Future columns are planed for windows, event management, menu bars, and a review of Prospero Pascal.

MORE

```
PROGRAM  Count;
{Written by Charles G. Adams, CompuLogic Development Corporation}
{(C) Copyright CompuLogic Development Corporation}
{Global Declarations}
   CONST
   {$I GEMconst.PAS}    {Include predefined GEM constants}

   TYPE
   {$I GEMType.PAS}     {Include predefined GEM Types}

   {$I GEMSUBS.PAS}     {Include predefined GEM Subroutines}

   VAR
   ExitVar : Integer;  {Variable used to exit Program}
   DataStr : String[255];   {Contains Input String}
   charcount,spacecount,commacount,periodcount : integer;
                        {Contains String Counts}

{The following GEM statements create a dialog box used for character
  input}

Procedure Input_Dialog;
{Procedure Specific Declarations}
Var  Input_Box : Dialog_Ptr;  {Define the Dialog Pointer}
     Instruct,Choice,Data_In,Help_But,End_But,Exit_But,Item_Color : Integer;
                  {Define the items within the Dialog Box}

     Begin {Input_Dialog}

          Input_Box := New_Dialog(7,0,0,40,15);
          Item_Color := D_Color(Black,Black,True,0,White);
          Data_In := Add_Ditem(Input_Box,G_Ftext,Editable,
                       10,2,20,1,0,Item_Color);
          Instruct :=
Add_Ditem(Input_Box,G_Text,None,3,5,34,1,0,Item_Color);
          Exit_But := Add_Ditem(Input_Box,G_Button,
                       Default|Selectable,Exit_Btn,15,10,10,3,1,Item_Color);
          Set_Dtext(Input_Box,Instruct,'Enter your text, Click on Exit',
                       System_Font,Te_Center);
          Set_Dtext(Input_Box,Exit_But,'Exit',System_Font,Te_Center);
          Set_DEdit(Input_Box,Data_In,'                    ',
                       'xxxxxxxxxxxxxxxxxxxx','',System_Font,Te_Center);
          Obj_SetState(Input_Box,Exit_But,Normal,False);
          Center_Dialog(Input_Box);
          Choice := Do_Dialog(Input_Box,Data_In);
          End_Dialog(Input_Box);
          Get_Dedit(Input_Box,Data_In,DataStr);
          Delete_Dialog(Input_Box);
     End; {Input Dialog}

Procedure Count_Char; {Procedure to evaluate the input string}
CONST
       Space = ' ';  {Define Space as a Constant}
       Comma = ',';  {Define Comma as a Constant}
       Period = '.'; {Define Period as a Constant}

VAR  J,K : Integer;

     Begin {Count_Char}
     charcount := 0;
     spacecount := 0;
     commacount := 0;
     periodcount := 0;

     K := Length(DataStr);   {Determine the length of the input
                                string}

          For J := 1 to K DO      {Looks at one character at a time}
             BEGIN
             charcount := charcount +1;   {Counts all characters}
             IF DataStr[J] = Space        {Is the character a Space}
                THEN spacecount := spacecount + 1  {Increment Space
                                                    counter}
             ELSE IF DataStr[J] = Comma   {Is the character a Comma}
                THEN commacount := commacount + 1  {Increment Comma
                                                    counter}
             ELSE IF DataStr[J] = Period  {Is the character a Period}
                THEN periodcount := periodcount + 1  {Increment Period
                                                      counter}
          END; {FOR}

     End; {Count Char}

Procedure Output_Dialog; {Dialog Box to display results of Count_Char}

VAR  Output_Box : Dialog_Ptr;
     Chars_Out,Spaces_Out,Commas_Out,Periods_Out,Choice,End_But : integer;
     CharStr,SpaceStr,CommaStr,PeriodStr : String[255];

     Begin {Output_Dialog}

          Count_Char;
          Output_Box := New_Dialog(7,0,0,40,20);
          Chars_Out := Add_Ditem(Output_Box,G_Text,None,
                       10,2,20,1,0,$1180);
          Spaces_Out := Add_Ditem(Output_Box,G_Text,None,
                       10,5,20,1,0,$1180);
          Commas_Out := Add_Ditem(Output_Box,G_Text,None,
                       10,8,20,1,0,$1180);
          Periods_Out := Add_Ditem(Output_Box,G_Text,None,
                       10,11,20,1,0,$1180);
          End_But := Add_Ditem(Output_Box,G_Button,
                       Default|Selectable|Exit_Btn, 15,15,10,3,1,$1180);
          WriteV(CharStr,charcount,' characters');
          WriteV(SpaceStr,spacecount,' spaces');
          WriteV(CommaStr,commacount,' commas');
          WriteV(PeriodStr,periodcount,' periods');
          Set_Dtext(Output_Box,Chars_Out,CharStr,System_Font,Te_Center);
          Set_Dtext(Output_Box,Spaces_Out,SpaceStr,System_Font,Te_Center);
          Set_Dtext(Output_Box,Commas_Out,CommaStr,System_Font,Te_Center);
          Set_Dtext(Output_Box,Periods_Out,PeriodStr,System_Font,Te_Center);

          Set_Dtext(Output_Box,End_But,'Continue',System_Font,Te_Center);
          Obj_SetState(Output_Box,End_But,Normal,False);
          Center_Dialog(Output_Box);
          Choice := Do_Dialog(Output_Box,0);
          End_Dialog(Output_Box);
             Delete_Dialog(Output_Box);
     End; {Output_Dialog}

Procedure Exit_Alert; {Tests if anymore strings are to be input}
     Begin {Exit_Alert}
          ExitVar := 1;  {Initialize to enter more strings}
          ExitVar := Do_Alert('[2][Enter another input string][ Yes | No
],0);
     End; {Exit Alert}

Begin {Main}
     If Init_Gem>=0 Then Begin
        Repeat
           Input_Dialog;
           Output_Dialog;
           Exit_Alert;
        UNTIL ExitVar = 2;
        Exit_GEM;          {Closes virtual workspace}
     End; {Init}
END. {Main}
END. {Count}
```

END



Menus, Menus and More Menus or Menus Cubed

by Eric Thornton

An easy to use, and attractive menu can turn a simple, text-only program into a primary use utility. In this issue I'll present four different menus for GFA basic programs. If you have a different basic then maybe only three of them will be convertable.

The primary function of a menu is to save the user from having to remember how to ask the program to switch between various functions. Another function is to allow the user to speed up his/her time at the keyboard and maybe save some typing and some time.

The first menu is a standard (GFA) GEM menu bar. It is the standard for GEM based machines. But first the program needs an initialization section.

Two arrays will be used in this test program. The first is M$(); it will hold the actual choices of the menu. The second, M2$() is used only for the GAM menu which requires a special format.

The data pointer is restored to point to the first data statement in the program. It may seem that this is wasteful, but I assure you it is a good idea. You never know when, in modifying a large program, you might accidentally add a data statement sequentially before an earlier one.

The Data consists of ten quoted strings which will be used in the formation of the menus. There is one extra space before and after the words in each string. This was done because the GEM menu looks better that way. Some other menus also look better with the options indented from the title and selection lines. Remember all menus should have an exit option.

The Gem Menu

So as to make room for possible desk accessories, the first few elements of the string array need to be filed in a specific format. The Quoted numbers "1", "2", etc, will never be seen. They are just placeholders, and are replaced with the desk accessory entries.

Each drop down needs to be followed by a single empty string and the final drop down needs a pair of empty strings. This is so GFA can tell where each menu title is and where the whole bar ends. I simply moved the menu data from M$() to M2$(), and then tacked on two place strings to show the end.

On Menu Gosub Gemmenuselect: tells GFA where the menu handler is, so when a menu event occurs, the program control is switched to the subroutine named.

To show a menu bar installed in a string array, you use the command "Menu M2$()". The rest of this menu routine is very simple. It just DOes a LOOP when waiting for a menu event. The command "On Menu" checks for a menu event. The "Exit If Item=20" leaves the loop if the item for QUIT is selected. To hide a menu bar

use the command "Menu Kill", named for obvious reasons. The description of the menu handler is found at the end of this article.

Numeric Keypress Menu

This menu lets the user select a numbered line of the menu to pick. Each option of the menu is printed out with a number and an equal sign before it. The another DO... LOOP takes over and positions the cursor below the lowest menu option, and waits for input.

The numerical value of the string entered is converted into a number, ("10" = 10). If the number is in the range one through nine, then an alert box displays which option was selected. If the option was 10 then we know that it is QUIT, so the loop is exited.

Mouse Line Menu

This menu is selected by the mouse being clicked while it is on the line where the option is printed. The menu is displayed as above; the numbers in this case being irrelevant (irreverent?). "Graphmode 3" says to GFA that when something is drawn in black on the screen and there is a white dot, then draw a black dot. Conversely, if there is a black dot there, then draw a white dot.

Remember our friend DO... LOOP? Well he's back. The mouse position and button state is read with the command "Mouse Smx,Smy,Buttn". The next two lines: "While Mousek0 Wend" wait for the mouse button to be released before going on.

The mouse X and Y positions are converted into character coordinates with a very simple formula. "Mx=Int(Smx/8)" which divides the screen coordinates into 8*8 squares and finds which one the mouse is in.

If the mouse-line is less than ten, then we know the mouse is on a line with a menu option. So we draw two solid boxes in black, one right after the other. But remember that we're in graphmode 3. What this means is that all the black in the box becomes white and visa versa. The second box command reverses the area again, and thus restores it to normal.

Now check to see if the button was pressed. If so then the mouse-line is increased by one. Check for the valid range. If the range is 1-9, then an alert box tells which option was selected. The DO... LOOP is exited if the selected option is 10 (QUIT). To be on the safe side the graphmode should always be restored to 1.

Rotation Menu

This menu shows only one option at a time, and then waits for the left button to tell it if this is the option the user wants. If the right button is pressed then the next option is shown, by looping around at the end to the start of the option list again.

The index of the first element of the menu string is 1, so we start with that value in INDX. Instructions are printed to the screen. DO... LOOP starts off again. Ya' just can't keep a good DO... LOOP down.

The line where the option will be printed is cleared by printing 40 spaces. This is because the option strings are different lengths. Then the correct (INDX) option is printed.

A WHILE... WEND loop waits for a button press, then another one waits for the button to be released. Otherwise, the options flip by so fast it is hard to stop on the one you want. If the button pressed is the right one (2), then the value of INDX is increased. It is a check to see if we have exceeded our maximum. If so then INDX is set to 1. This makes the next option after 10 be 1.

ELSE means that if the right button was not pressed, (we wouldn't have gotten here if SOME button hadn't been pressed) then it must be the left button. So clear a line, and show which option was selected. Finally, if the button pressed was 1 (left) and the option selected was 10 then exit the loop.

The One and Only Subroutine:

Procedure Gemmenuselect: picks up the number of the item selected (GEM menu) from the built in array MENU(), and uses an alert box to display the selected string. "MENU OFF" turns this dropdown TITLE back to black on white, from its reversed state.

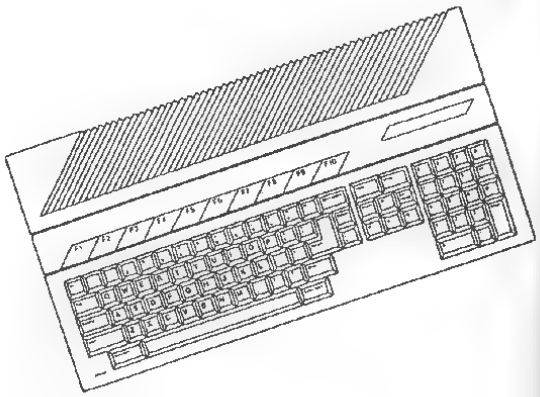
Final Notes:

The reasons for skipping the first element of the array are:

- 1, because the first line number in a PRINT AT() statement is 1.
- 2, I usually use a On A Gosub sub1,sub2,sub3... to branch after menu selection, and if the value of A is zero nothing happens.
- 3, I like it that way, it gives me a spare string later on for menu Title (not used here) or such like.

Did ja have a good time? I did! Until next time remember to eat the meal not the menu.

MORE...

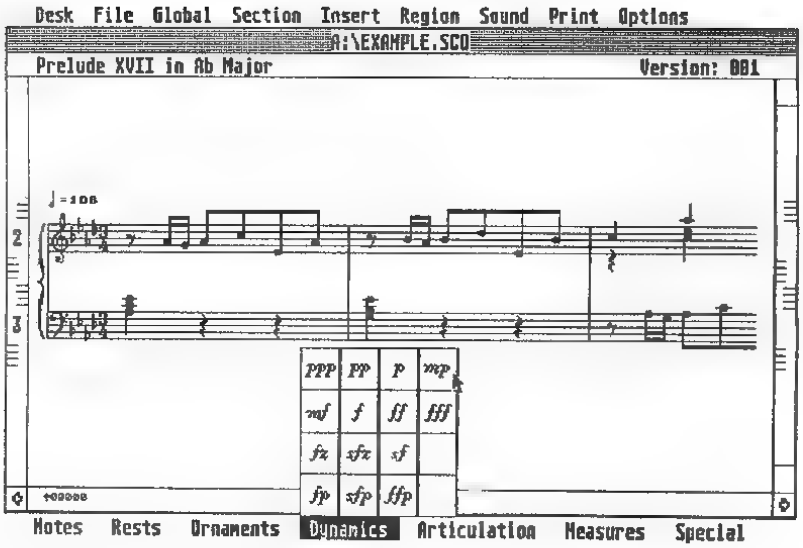


Menus, Menus, and More Menus

```
' Menus, menus and more menus - or - m^4
' For ST Applications by Eric Thornton
Dim M$(10),M2$(23)
Restore Dat
For X%=1 To 10
  Read M$(X%)
Next X%
Dat:
Data " Option 1 "," Option 2 "," Option 3 "," Option 4 "," Option 5 "
Data " Selection 1 "," Selection 2 "," Selection 3 "," Selection 4 "," QUIT "
'
' GEM MENU
Menu1:
M2$(0)="Desk"
M2$(1)="  About...            "
M2$(2)="---------------  ----'
M2$(3)="1"
M2$(4)="2"
M2$(5)="3"
M2$(6)="4"
M2$(7)="5"
M2$(8)="6"
M2$(9)=""
M2$(10)=" OPTS"
For X=11 To 11+9
  M2$(X)=M$(X-10)
Next X
M2$(22)=""
M2$(23)=""
On Menu  Gosub Gemmenuselect
Menu M2$()
■
  On Menu
  Exit If Item=20
Loop
Menu Kill
Cls
'
' numeric keypress menu
Menu2:
Cls
For X=1 To 10
  Print X;"= ";M$(X)
Next X
Do
  Print At(1,12);
  Input ";",A$
  A=Val(A$)
  If A>0 And A<10
    Alert 1,"KEYPRESS MENU "+M$(A)+" "+Str$(A) 1,"OK",Ret
  Endif
  Exit If A=10
Loop
Cls
'
' mouse line menu
Menu3:
Cls
For X=1 To 10
  Print X;"= ";M$(X)
Next X
Graphmode 3
Do
  Mouse Smx,Smy,Buttn
  While Mousek<>0
  Wend
  Mx=Int(Smx/8)
  My=Int(Smy/8)
  If My<10
    Pbox 0,My*8-1,15*8,My*8+2
    Pbox 0,My*8-1,15*8,My*8+7
  Endif
  If Buttn=1
    A=My+1
    If A>0 And A<10
      Alert 1,"KEYPRESS MENU "+M$(A)+" "+Str$(A),1,"OK",Ret
    Endif
  Endif
  Exit If A=10
Loop
Graphmode 1
Cls

  rotation menu
Menu4:
Indx=1
Print At(1,10);"LEFT Button selects this option"
Print "RIGHT button = Next option"
Do
  Print At(1,1);String$(40," ");
  Print At(1,1);M$(Indx)
  Buttn=0
  While Buttn=0
    Buttn=Mousek
  Wend
  While Mousek<>0
  Wend
  If Buttn=2
    Inc Indx
    If Indx>10
      Indx=1
    Endif
  Else
    If Indx<10
      Print At(1,8);String$(40," ");
      Print At(1,9);M$(Indx);" ";Str$(Indx)
    Endif
  Endif
  Exit If Buttn=1 And Indx=10
Loop

Procedure Gemmenuselect
  Item=Menu(0)
  Alert 1,"MENU SELECTION|"+M2$(Item)+" "+Str$(Item),1,"OK",Ret
  Menu Off
Return
```

END

Midi Beat



EZ Score Plus

Hybrid Arts
11920 W. Olympic Blvd.
Los Angeles, CA 90064
213/826-3777
BBS 213/826-4288
$150

Type: Music Scoring
Level: Beginner to Intermediate
Features: Transcribes tracks from Hybrid Arts sequencers into traditional music notation; also allows creation of sheet music from the computer keyboard, mouse and MIDI input; easy to use GEM user interface; lyric editor; guitar chord symbols.

Introduction

If writers and editors can use the ST for desktop publishing, why can't musicians use it for desktop music publishing? Why can't we use the computer to print out music scores and parts in better handwriting than, say, mine?

You're right, it was a trick question. We can. Scoring software for the ST is one of the last areas of MIDI software to catch up, and now we are seeing some choices. Unfortunately, none of the ST scoring software has yet achieved the power available for either the Mac or MS-DOS machines.

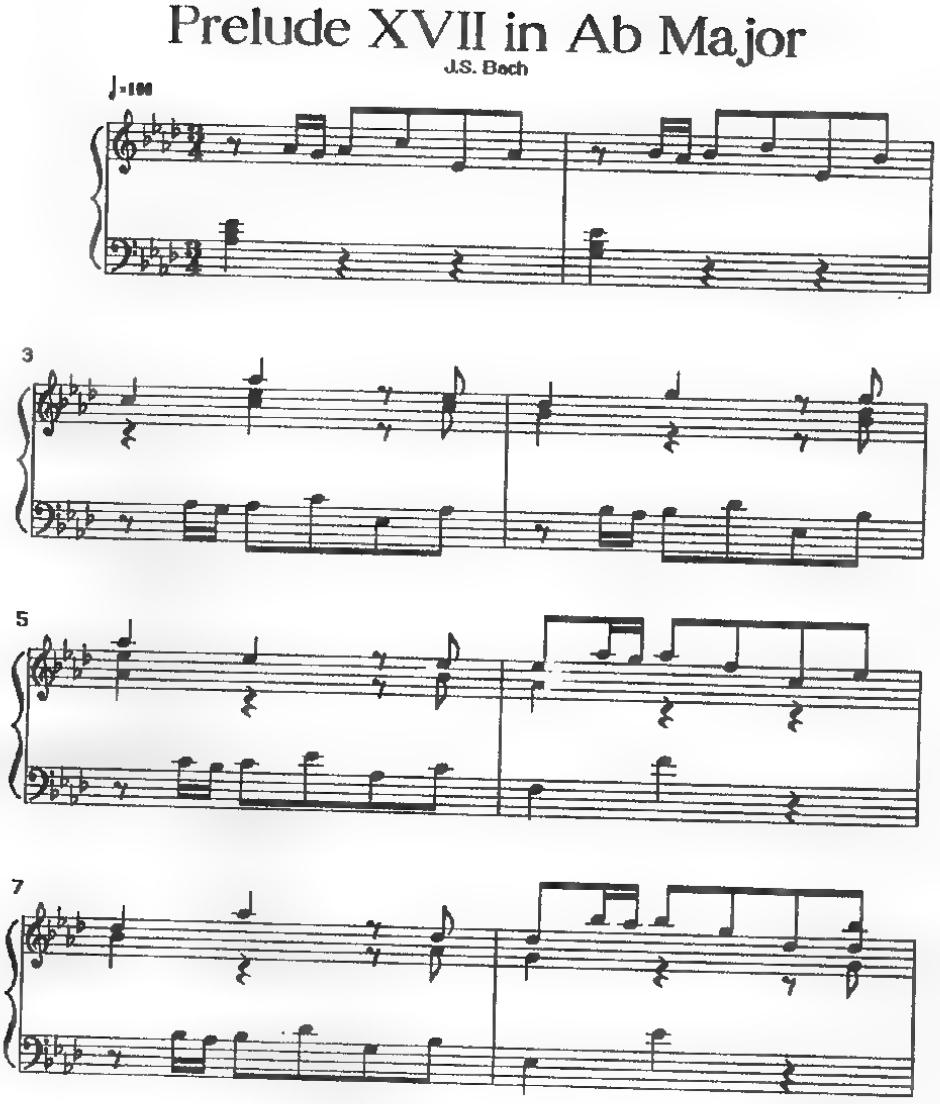
The Current Situation

The Copyist (three levels at $100, $250, and $450) from Dr. T's prints BEAUTIFULLY. The dot matrix driver is excellent; more money gets you plotter and laser support; and it is quite flexible and feature packed. Unfortunately, it's unnecessarily difficult to work with. Quiet Lion's affordable Mu-Script I ($100) includes a basic sequencer, but it's inflexible and the printed output is poor.

Certainly the ST is capable of both an easy and flexible user interface and professional quality results so there is hope. Sonus has said Superscore ($300, with built in sequencing), is shipping, but it hasn't shown up yet. Steinberg/Jones is working on Masterscore ($350), and their Pro24 III ($295) sequencing program has integral notation editing/printing. Compu-Mates is working on FinalScore ST ($200) and Hybrid Arts is just releasing EZ Score Plus 1.1 ($150).

EZ Score Plus

I picked up EZ Score Plus 1.0 from Hybrid at a music trade show in mid January. Hybrid is having problems with a new disk based copy protection scheme, and after I tried to make a backup, the disk would not work. Thanks to the local distributer I got another disk to try. This one worked, though I didn't want to risk trying to back it up. The manual encourages making one



This is an example page printed in the "final" mode without using the overstrike option.

backup, though it says the program must be run from the original disk. I get really nervous about applications software, and I like to be able to make useable backups.

If you're a pirate, I blame YOU for this copy protection mania! Here's to companies like Soft Logik, Michtron and Time Works who are brave enough to trust us. If you abuse that trust, may you be chopped into moderately sized bits and used as shark bait.

Aside from the copy protection problems, (which Hybrid blames on the duplicator), version 1.0 was an encouraging start. Just before this reviewer's deadline, programmer Tom Bajoras was kind enough to get back to me and send a beta version of 1.1. Hold the presses! 1.1 is even more worth talking about, and Tom seems pretty darn sure it will be out by the time you read this.

Where 1.0's printing was barely adequate, 1.1's printing is quite respectable. You can choose between rough, draft, and final printing (dot matrix only), and a double strike option is also available. The result is excellent looking sheet music.

EZ Score Plus has a lot of very nice features. You can easily transcribe individual tracks from Hybrid's EZ Track, Sync Track and SMPTE Track sequencers, (I used and was impressed with the latter). In addition to GEM drop down menus, there are pop UP menus for notes and accidentals, rests, ornaments, dynamic markings, articulation markings, measure symbols, and special symbols (including chord types and even guitar chord symbols). Both drop down and pop up menus can be made to activate when touched or when clicked on. Other thoughtful features include auto beaming, auto stem direction (both up and down beamed stems are available for putting two parts on one staff), a nice lyric editor which lines up syllables with notes, the formatting of disks from within the program, the ability to save a screen in Degas format, auto rest and tie minimizing options, and automatic quantizing when importing a track from any Hybrid Arts sequencer file.

The program supports input from the mouse, computer keyboard and/or MIDI keyboard in any combination. You can use the mouse to grab a symbol from a pop up menu and put it on the staff where you want it. Once placed it unfortunately cannot be picked up again and moved, though clicking again on a note, if you haven't selected a different symbol, will delete it.

A fast way to work is to use the numeric keypad to change the value of a note or rest, while using the mouse to place the symbol on the score. A clever MIDI implementation allows you to play notes onto the score from a MIDI keyboard: using the mod wheel to change the note value, the pitch wheel to move the symbol forward or back, and a MIDI switch (such as the sustain pedal) to toggle between notes and rests. This would be easier and faster if the program had an option to automatically space notes and rests correctly on the score as you "typed" them in from the MIDI keyboard.

Problems include the inability to pick up and move symbols once placed on the score, (though this is made up somewhat by the toggling nature of notes), the lack of a function to automatically format notes and rests into proper horizontal position in a measure, and a three stave limit. Also there is the lack of any way to separate out and print staves 2 & 3 individually in a multi-stave score and an inability to slant beams. There is no preview on screen of what a page layout will look like although there is a menu which lets you decide where titles, page numbers and such will be. Some small accuracy problems occasionally surface when transcribing tracks from SMPTE Track. There is a lack of the ability to save scores in a format readable by a sequencer. Finally, there is the copy protection problem.

Hybrid Arts is talking about putting dongle protection on the parallel port. It would have a pass-through to allow unimpeded printer hook-up, and it would be sold separately. It would work for multiple Hybrid programs. If the dongle would allow SMPTE Track to run without the SMPTE box being hooked up, this would be appreciated by those who use SMPTE Track for live performances as the box is an extra hassle on stage.

As I understand it, EZ Score Plus is an evolving project. The goal is both an easy-to-use package (EZ Score Plus) and a professional level 60 stave music scoring package (MIDI Score). As the bugs are fixed and features added to EZ Score Plus, updates will be available (for $15 a shot). EZ Score Plus owners should be able to upgrade to MIDI Score for the difference in price between the two programs. MIDI Score probably won't be released before fall, and its price has not been announced.

Conclusions

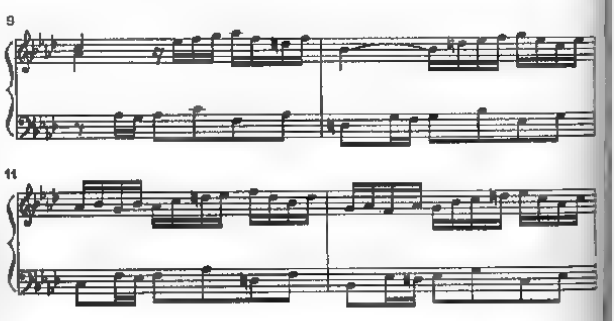
Hybrid Arts has combined a decent user interface with good looking printed output. This is something many people have been waiting for. The input combinations are powerful; the editing is easy; symbol placement is flexible; and the manual is good. It's an impressive first effort. The program is handy for printing parts from sequences and for creating music for piano and a solo instrument or voice. The lyric editor is a great feature.

EZ Track Plus is not bug free, but updates will be made available. The three stave limitation does restrict its usefulness.

Transcribing parts from sequences works now only if you own a Hybrid Arts sequencer, though there should be a utility available from Hybrid sometime to translate between the Hybrid file format and the new MIDI file format standard. Utilities should be available from other companies as well, which will allow anyone's scoring program to print tracks from anyone's sequencing software.

In conclusion, the future looks bright for desktop music publishing on the ST. EZ Score Plus is a big step in the right direction.

Reviewed by Jamie Krutz



dBMAN™

The intelligent database manager

Review Board

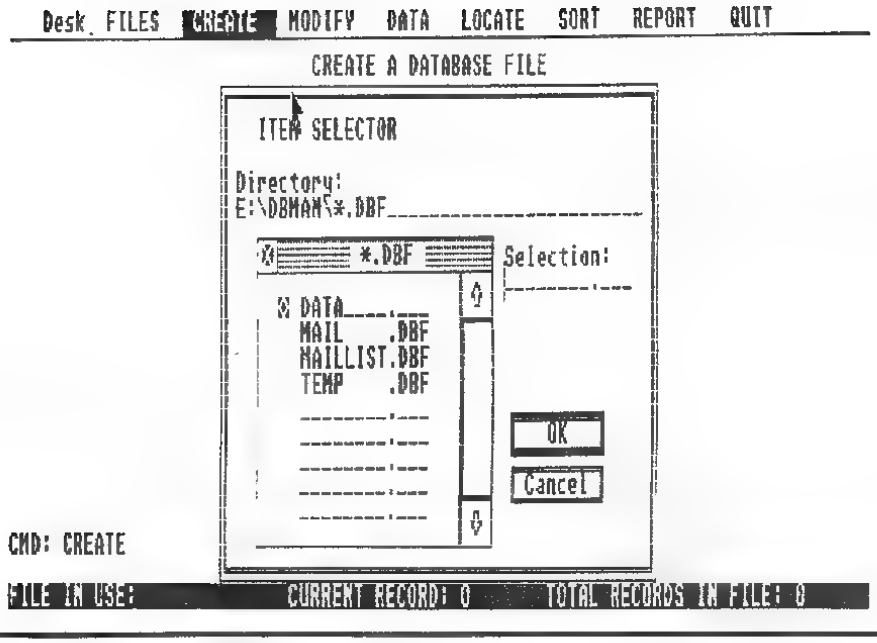
VERSASOFT
4340 Almaden Expressway Suite 250
San Jose, CA 95118
(408) 723-9044
$249.95

DBMAN is a product that will accept most Dbase commands. It allows the user the environment to produce relational data base files and programs. Version 3.01 has a program called **ASSIST** that will help you use **DBMAN** commands. **DBMAN** has a run-time package available, so stand-alone applications are available. The **DBMAN** source code can be speeded up through the use of a compiler program. Overall, **DBMAN** is a extremely useful product.

Most people who have used the IBM pc are familiar with the Ashton Tate product 'Dbase III plus'. It allows application programs to be developed by computer users who may not be familiar with other programming languages. When asked to review **DBMAN**, I immediately compared it with Dbase. I find that Dbase has some features I miss in **DBMAN**. Dbase has the use of an up arrow key to call back previous commands; on **DBMAN** the F three key will call back only the last command. I also like the config.db file associated with Dbase and miss not having the ability to program my function keys. **DBMAN** keeps stride with Dbase on all other topics. The ease of setting up a working database is unbelievable. If you have Dbase experience, this product will be working for you in very few keystrokes. I have written a few Dbase applications, and have had very little trouble using my pre-existing code in **DBMAN**.

Powerful relational commands exist; through the use of procedure files custom environments are easily produced. In creating an accounting application, it is easy to create the database files used and program menus needed to accomplish tasks needed. All information is first broken down into permanent and transaction files. Once a dbf file is used, all data elements stored are open for use. If more then one program is called, the variables from a preceding file can be accessed through alias headers such as X.cash or Y.cash (where x is accessible to y). Custom input screens are obtainable through the use of @ say and @ get commands. There are many other commands for creating interesting displays. Pmenu commands are used to create GEM type input graphics, so it is really easy to use GEM in your application.

ASSIST is a program file that will help you in many ways. **ASSIST** tries to be to **DBMAN** what assist is to the Dbase version. I feel that it does a good job of assisting me through the mundane chores of creating database files and testing index files. The real beauty of **ASSIST** is that it uses most of the commands that **DBMAN** has to offer. If you are interested in writing



your own menus and open files from a program file, there are some good examples in the ASSIST program that will be extremely beneficial to you.

DBMAN has available a run-time package that allows users to create custom applications and then tokenize the commands into a program that can be distributed for lease, sale, or for use on other systems. It needs the run-time program supplied with the package to use the programs produced from source code. This is a nice feature if you plan to market or have multiple users of any application programs you are developing.

Speed and size are some advantages of compiling software. VersaSoft offers a separate compiler for **DBMAN** database applications. The compiler normally sells for $249.95, but is available at $99.00 for a limited time.

After purchasing **DBMAN** for my ATARI 520 home computer, I had a tremendous feeling of delight. After all, here is most of what makes Dbase great right here on my own Atari computer. I had no experience with Dbase when I first purchased **DBMAN** for my Atari. All I knew is that I wanted to keep track of the books for my small business. I was used to a database program on the Commodore 64 that used a main menu to access the program commands. Upon first loading **DBMAN** I relized that large amounts of time were necessary to learn all the commands. The command prompt, sitting in front of me, had the key to great power if I could learn enough to unlock it. After a few months of reading, the prompt was a friend who is seen after creating the code to be used in a text editor. The documentation is great and most manuals for Dbase can be used on **DBMAN**. As is the case with most data base programs the editor supplied needs a lot of help. I still chose to use 1st Word. I had to fix a bug as the mouse would disappear. I wrote a ML program to turn on the mouse, before calling my word proccessor, and another to turn it off before returning to **DBMAN**. These programs are called from inside my command file, I will include them for reference and use, i.e.:

```
RUN C:\MOUSE_ON.PRG
RUN C:\1st_word.PRG
RUN C:\MOUSE_OF.PRG).
```

DBMAN has fixed this bug in later versions. **DBMAN** is the perfect set of tools for anyone who wants to create a custom application. I have not used the built-in report generator but choose to create my own reports using the command language. If you are experienced in Basic, you will find the commands easy to use. I have seen commercial applications achieved through extremely ingenious programming. The product I like best is a video store management end product program for point of sale called "The Video Man" by Steve Yeny of Jazzman Software (available through Intersect Software 1-800-826-0130 toll free or in Florida at 1-813-923-8774). This program uses ram based drives and relational databases. Completly graphic, it showed me that **DBMAN** is capbable of using GEM to create easy-to-use full featured applications. Everything that makes the ATARI 520 an easy-to-use computer is available through **DBMAN**.

Reviewed by Joel Fradkin

OIDS

Software Heaven
PO Box 112489
San Diego, CA
$32.00 ?

Many years ago, there was an extremely popular game called Protector, where you were operating a helicopter in some small nation where your job was to rescue the American nationals, and carry them back to the American Embassy. **OIDS** uses the same theme, but has taken this simple idea to the extreme and made this simple game into a memorable experience.

The thesis of the game is that the evil Biocrets have created a race of android slaves called **OIDS** to work for them. Their life is full of misery and you have so much compassion in your heart that you are willing to fly your small fighter to one of the many Biocrete planetoids. There, you must blast open the factories with only one blast of your nuclear pellet, so as not to destroy the inhabitants, land and rescue the **OIDS**. Once you have captured eight **OIDS**, you can return to your mother ship and the **OIDS** will have a life of ease, with no possibilities of servile labor and all the oil they need to maintain their sustenance.

The game play is fairly complex and sophisticated. Your space craft can rotate in 22.5 degree increments as you shift the Joystick to the left or right. When you pull up, your trusters fire and when you pull down, a shield appears that will protect you from the various weaponry and hard landings. Your shield will be operational only for a short time before it needs to be recharged. The asteroid has been set up so that it is dynamically realistic. Gravity starts to tug at your craft, and you need to fire your rockets to move in any direction or to overcome the pull of gravity. Your spacecraft is not indestructible and landings have to be rather gentle so that the ship survives. Once you have rescued the **OIDS** on the planetoid, you can return to the mothership and it will take you to another mission. The disk comes with five galaxies that have four to six different parts to each.

Software Heaven (FTL) is known for their graphics and this game is no exception. While the ship is not very detailed, it moves smoothly and gracefully around the asteroid. The Joystick commands are also well thought out and the little **OIDS** are cute. The various enemy objects are colorful and shimmer as well as pulse when such actions are needed to make the game more colorful. The most imaginative part of the game is the terrain which you must fly up, around, and through to destroy all of the ships and objects. Of course the high scores are saved to the disk for posterity. Each of the various galaxies are individually saved.

If this were all the **OIDS** was, then it would be an above average shootemup with good graphics and about 20 different screens of various complexity. There are some additional factors that must be taken into account. First of all, this game can be played from the keyboard as well as the Joystick. Big deal you might say. However, you can configure the keyboard to follow the commands that you want so that the warped logic of your mind will prevail rather than the warped logic of the designer.

But the high point of the game is the game editor. You can create your own planetoids and galaxies. From the game editor menu, you can select which one of the existing planetoids you wish to modify, or you can start from scratch. Once you start, there is a cursor which can be either one of the eight different mountain shapes or one of the bases of the Biocretes. The **INSERT** keys is used to change the selection while the arrows keys move the piece around to create your planetoid. It is here that you see the involved concepts that have made up the games. (While you are actually playing it is difficult to admire a vessel that is trying to destroy you.) The bases for the Biocrets range from the Helljet, Saucer, Guard and Burst to the Inducer, the reaper, the Decroids. You of course can have the typical guns, launchers, Gravitods and home bases which are peculiar to all planetoids. As an extra bonus, you can scatter teleports to help in your survival.

Once you have designed a good planetoid, you can save it to disk for future play. To help perfect the Planetoid, there are a number of extra commands in the menu. You can use the **'help'** key to align your mountain to the closest mountain to make smooth looking graphics. You can see a 1/8 scale view of the entire planetoid. You can press **'P'** to start up the game for playtest with a ship at your current cursor position. Pressing 'Q' allows you to quit the testing and resume the editing. The **'M'** key lets you start up the game where the mother ship drops you off. You can also restore all of the ground bases which were destroyed during play testing. Finally you can adjust the alien difficulty and general difficulty for the planetoid that you are generating. Lastly, there is the contest for the best **OIDS** galaxies. FTL will award $100 every month for the best one, and the contest is valid until March 1989.

Overall, **OIDS** uses a good concept for an arcade game. It also gives you a large measure of control for designing new and interesting galaxies. Best of all, it is fun to play.

Reviewed by Sol Guber

ST PICTASCAN

Software by Paul Franson & Kurt Casby
Hardware by Orrin McGill
E. Arthur Brown Company
3404 Pawnee Drive
Alexandria, MN 56308
(612) 762-8847
$149.95

E. Arthur Brown, for those of you who have never received one of his "Computer Enthusiast" catalog/newsletters, is to Atari computers what Drew Alan Kaplan, of the DAK catalog, is to stereo and related electronics. He is an irrepressable gadfly who writes all of his ad copy and provides *irresistable* deals on software and other products related to Ataris, PCs, and most recently, Amigas. His is not strictly a mail order only business, so there is a chance that your local dealer is carrying PICTASCAN. If not, tell your dealer to start stocking this scanner because it is a very good, inexpensive scanner fit for use in desktop publishing projects and for other tasks in which you need digitized pictures. Like IMG Scan, PICTASCAN can be attached to the printer head of any dot matrix printer, turning it into a sheetfed scanner. Unlike IMG Scan, it is an LED optical sensor in a protected, metal box that comes with three mounting brackets and plugs into the second joystick port of your ST or Mega, rather than the cartridge port. While I have no knowledge of the hardware specs, from the results I've seen, I cannot disagree with their claim that the joystick port offers higher samplings (4,000 to 6,000 per line) than the cartridge slot.

While they've made mounting the scanner simpler, I've got to admit I have spent a great deal of time working out the ideal setup for my SG 10. But lately, I've had less trouble and have been able to spend more time actually doing scans. The problem was in getting the sensor properly positioned (to touch the page I wanted scanned) between the bar that holds down the paper and the brackets of the tractor feeder, so when making full passes it didn't hit the brackets or the rollers on the bar. Yes, they recommend leaving the tractor feeder attached and I agree, despite the hassles, because having another sheet tautly placed under the page you want to scan ensures a smooth, consistent pass across the page. [WARNING: Do not for any reason use an original you cannot afford to have damaged. PICTASCAN does leave "teeth" marks where it makes returns. Use a photocopy.]

PICTASCAN also comes with a program to capture and save your scan, but for the most part that is all it does. You do not make many adjustments of the scan's quality with this software except for magnification and cropping of the image. If you need to adjust the contrast, colors, or area of the scan, you must adjust the scanner unit itself, which has two small, recessed screws on the top of its box. One screw handles the stippling effect and is recommended for use with color images to get the right contrast and texture. The other basically handles the black and white contrasts. You will need an eyeglass-type screwdriver for these adjustments. I suppose there are techies who have such thin-sized screwdrivers laying all over their homes, but it would have been nice if Brown & Co. had included such a screwdriver with this package.

So again, unlike IMG Scan whose software can seemingly take a completely white or black captured screen and turn it into an image through its gray scale adjustments, PICTASCAN requires that you get the perfect image onscreen through adjusting the scanner before the software comes into play. This is a process of trial and error that can take awhile when just starting out. The only exception to this is with the way you configure the printer for the scan. The software comes with configuration files for scanning the center "half" page, left and right halves, and the full page at 3/216ths and 1/216th linefeeds, using codes for Epson compatibles. (Oddly, I found for the first time that I did not have to switch my DIP toggles to Epson/IBM mode for my Star printer to use these codes.) There is an Create Config function for refining these commands for your own printer and for making further modifications of the pass length. This fiddling with printer codes may be desirable, since the first image you get may be very condensed and may not be complete. You must adjust the printer codes so the image is not cut off by the scanner. However, with the half-page codes provided, you may be able to position your image page on the printer to effectively use these config options.

Halftones using Pictascan

Three basic dialogue boxes comprise the software interface with the scanner. The first one concerns the initial parameters for loading raw data, creating or loading the config files, or beginning the scan. You can hit Return to begin the scan, which is a convenient touch

I put a frame around this first image from PICTASCAN to show how it looked after it was scanned. The image at the right is the adjusted image and is more in proportion to the original TIME cover that I photocopied for the scan. The text in the thought balloon is still pretty hazy, but Superman, the headline and the rayed lines are in great shape. The third image (at right) is a blowup of his face, and this is impressive–try blowing something up using DEGAS ELITE and you'll see what I mean. This ability to adjust size without deterioration of your image is a very powerful feature of the software that comes with PICTASCAN.

since this is an operation where you're not seated at the computer but most likely hovering between the printer and the computer. To pause the scan, you press "p" as the scanner is passing to the right (unless you are attempting a reversed scan -- it's possible!). Then you come to the next dialogue, which contains buttons for Abort, Done, and Continue, with the latter button being the Return default. The third box handles the completed scan. While not as exotic as the IMG Scan software, this is a powerful and useful adjunct to the scanner.

The middle part of this third box contains information on the number of lines scanned and average number of samples per line. Below this information are text features that can be used to edit the scan and may become useful as you learn more about the mechanics of scanning and editing. As the docs state, "using the mouse is easier, faster, and more fun!" I agree, but it's nice to have the means for precision if the need arises. Pay attention to this info every time you do a scan and you may learn something. The rest of the box consists of ten buttons for Deleting files, Formatting disks, Quitting to the previous dialogue (where you can exit the program), Saving the picture as a DEGAS file or Raw Data file, and for manipulating the data.

What you have here is a two-buffer setup, one that holds a permanent view of the scan (accessed by clicking on VIEW RAW) until you scan again; and the other that is volatile (accessed by clicking on VIEW or SET AREA), where you create the image you will save as a DEGAS file. When you click on the View Raw button, you view the scan you originally saw as it was scanned, but expanded to fill the screen. This may or may not be in proper proportion to the actual image. Your cursor will be an arrow with an "R" attached to it to identify the mode you are in. By double-clicking and holding the left button down you can rubberband the area you want to transfer to the volatile View screen. This can be a tricky procedure and if done in either top corner of the screen can call up the Delete or Format command instead of the rubberband effect! Once you've indicated the area of the scan you want to work with, release the mouse button and you are transferred to the View screen where the image is redrawn to fit the screen or an area you set beforehand by clicking on the SET AREA button.

Thank goodness the redraws are quick, because this is an area where trial and error take over if you want to get the proportions properly captured. You can also do similar rubberband captures in the View buffer (the cursor is an arrow with a "V" attached to it), but if the area you set is smaller than the original image on that screen, part of the original stays. This area can be cleared by using the Erase button first. But an advantage of this seeming oversight is the ability to "build" an image from two scans, since only the Raw View buffer is cleared when a new scan takes place. Redraws can be aborted by pressing "a" and last changes can be eliminated by pressing Undo. What is truly remarkable with this software is that redraws are pixel perfect no matter what the size of the redraw. You can focus on any small detail of your scan, capture it, and have it blown up to screen size without loss, or rearrangement, of data. I wish it was possible to reload DEGAS files into this buffer, since even DEGAS Elite is incapable of resizing without unduly damaging the image.

As you can see from the samples included with this review, PICTASCAN works very well with finely detailed line art -- much better than IMG Scan, and if not completely perfect, at least good enough to be worth porting to DEGAS Elite for cleanup. Keep in mind that these samples were generated with a color system and that a high-rez system would yield better results in its raw format, which would make the .PI3 file equally improved. If this is unsatisfactory, then your only alternative at this stage in the game is to buy the Navarone ST scanner at nearly ten times the price. I think you'll find that PICTASCAN delivers excellent scans across the full range of image types that you'll want to digitize for use with the most popular ST drawing and paint programs.

The docs are complete save for the need for more troubleshooting tips where it concerns making adjustments to the scanner unit. Instead of an index, the docs carry a complete reference section and diagrams of all dialogue boxes with page numbers and arrows indicating where in the docs information can be found about buttons and functions. The manual is written in the same breezy, almost-smug style that Brown's catalog is written. Also included are instruction sheets for setting up the mounts or creating a mounting brakcet for your printer.

While PICTASCAN, as is, is a satisfactory package, I would like to see more tools added to the software. For instance, for editing the captured picture, or the ability to reload DEGAS files once they've been edited if only for the software's excellent ability to resize and reshape images without distorting the data. If this feature was added, the software would be boon in and of itself to artists and desktop publishers. The raw data files are not usable by any other program that I've run across. I would assume that its data structure would yield a superior image to a DEGAS file, yet used with Easy Draw's Supercharger, you can get an excellent IMG file for use with Easy Draw, Timeworks Desktop Publisher, and the forthcoming Publishing Partner Professional. For Desk Top Publishing on a budget, PICTASCAN gets my recommendation for digitizing clip art and photos.

Reviewed by Donavan Vicha

Combining images to create a montage using Pictascan

Universal Item Selector

Applications & Design Software
A Part of Mac Donald Assoc.
226 NW 'F' Street
Grants Pass, OR 97526
(503) 476-0071
$15.95

Wow! What a great little utility. Why didn't Atari come up with something like this when they designed the ST operating system in the first place. What am I talking about? **The Universal Item Selector** by *Applications & Design Software* for $15.95.

Big deal, you might say. I already have an item selector, the one that comes with the ST operating system. Well not like this you don't. As you can see from the screen dump illustrations there are quite a few new options to this item selector. You can change drives with the click of the mouse, instead of having to type it in the new drive name. This is fantastic. I have a hard drive partitioned into four drives plus my disk drive. Now I can zoom back and forth between drives with the click of a mouse. You can rename, copy, delete, and even move files and folders (the old one is automatically erased). The number of files shown at one time is increased, and in addition to being able to scroll up and down you can scroll sideways to see how many bytes long the file is and its date. By clicking on the question mark in the right corner you can find out how many bytes are in a folder or the entire drive with the free disk space shown as well. Wild-cards can also be used to find, move, lock, unlock, and delete files.

By clicking on the **P** in the lower left hand corner you can get a hard copy of your directory on the printer. If this was not enough, as you can see from the second illustration, you can format disks getting up to 818K with a double sided disk with 10 sectors per tract and 82 tracts, twister mode is also supported to speed up disk reading and writing.

The program is used by putting it in an **AUTO** folder and is loaded into memory on boot up. There is also a desk accessary that allows the **Universal Item Selector** to be called from the desk top.

But what is particularly valuable about this program is that all of these functions are available from within any application that normally has an item selector or allows for desk accessaries.

Reviewed by Ron Schaeffer

New Software Etcetera

Should you have a product available for the ST, please send us a copy and we'll mention it in this column.
* *indicates item currently available*

Alpha Systems
1012 Skyland Drive
Macedonia, OH 44056
(216) 467-5665

* **Switch Back** ($70) a hardware/software package that lets you swap between programs, back-up programs, or flip back to a previous point in any program, and more.

Artisan Software
P.O. Box 3213
Fontana, CA 92334

* **Word Quest** ($18) program designed for hidden word game zealots, targeted for those who wish to add a proven entertainment feature to their periodicals.

Datasoft
19808 Nordhoff Place
Chatsworth, CA 91311
(818) 886-5922

* **Cosmic Relief** ($35) a humourous, multi-level animated action/adventure. Find Professor Renegade, the only one man who can stop an asteroid heading toward Earth. Tea cups, unicycles, and flying vacuum cleaners (?) are required for the trek.

Electronic Arts
1820 Gateway Drive
San Mateo, CA 94404
(415) 571-7171

* **Rockford** ($30) accompany the world's greatest archeaologist, Rockford, to five exciting locations. Avoid hazards such as falling rocks, fearsome fish, and...poisonous pizzas..(?)..

Mindscape Inc.
3444 Dundee Road
Northbrook, IL 60062
(312) 480-7667

* **A Guide to Selecting Educational Software** (Free) a leaflet that outlines the basics of software evaluation and selection. (Send a self-addressed, stamped business size envelope.)

The Sterling Connection
Box 4519
Berkeley, CA 94704

* **Templicity** ($25) a collection of 53 templates for both novice and experienced spreadsheet users. Available now for EZ Calc, SwiftCalc, Analyze!, Logistik, MasterPlan, and all verions of V.I.P.

Strategic Simulations, Inc.
1046 N. Regstorff Avenue
Mountain View, CA 94043
(415) 946-1353

Heroes of the Lance (available Summer 1988) joystick controlled single-player action game set in the *DragonLance* game world.

subLOGIC
501 Kenyon Rd.
Champaign, IL 61820

* **"Western European Tour" Scenery Disk** ($25) detailed scenery covering southern Great Britain, northern France, and southern West Germany. For visual and/or instrument flight.

MIDI Software Etcetera

Passport Designs, Inc.
625 Miramontes Street
Half Moon Bay, CA 94019
(415) 726-0280

* **Master Tracks Pro 3.0** ($395) provides 64 recording tracks for each sequence, and now also supports two independent MIDI outputs for a total of 32 channels, and more.

The reviews are in . . .

"'A Best Buy' I'm impressed"
David H. Ahl, Atari Explorer, Nov-Dec 1987

"If you've got an Atari, you probably need this program."
Jerry Pournell, Byte Magazine, October 1987

"pc-ditto is a winner."
Charlie Young, ST World, July 1987

"This is the product we have been looking for."
Donna Wesolowski, ST Informer, August 1987

"This truly incredible software emulator really works."
Mike Gibbons, Current Notes, September 1987

NOW! RUN THESE IBM PROGRAMS ON YOUR ATARI ST.

Lotus 1-2-3	Flight Simulator	Framework	Symphony
Enable	Ability	DESQview	Q&A
Sidekick	Superkey	Norton Utilites	dBase II,III,III+
Crosstalk IV	Carbon Copy	Chart-Master	Print Shop
EasyCAD	DAC Easy Accounting	BPI Accounting	Turbo Pascal
GW Basic	Managing Your Money	Silvia Porter's	pfs:Professional File

And Hundreds More!

---

pc-ditto is a software-only utility which expands the power of your Atari ST to imitate an IBM PC XT. No extra hardware is required (an optional 5.25-inch drive may be required for 5.25-inch disks). All your IBM disks will work "out-of-the-box".

pc-ditto features include:

- o both the 520ST and the 1040ST supported
- o up to 703K usable memory (1040ST)
- o not copy-protected -- installable on hard disk
- o imitates IBM monochrome and IBM color graphics adapters
- o access to hard disk, if hard disk used
- o optionally boots DOS from hard disk
- o parallel and serial ports fully supported
- o supports 3.5-inch 720K format and 360K single-sided formats
- o supports optional 5.25-inch 40-track drives

System requirements:

- o IBM PC-DOS or Compaq MS-DOS version 3.2 or above recommended
- o optional 5.25-inch drive is required to use 5.25-inch disks
- o 3.5-inch 720K DOS disks require a double-sided drive (Atari SF314 or equivalent)

See pc-ditto today at an Atari dealer near you, or write for free information!

$89.95

pc-ditto
by
Avant-Garde Systems
381 Pablo Point Drive
Jacksonville, Florida 32225
(904) 221-2904

Avant-Garde Systems, 381 Pablo Point Dr. Jacksonville, Florida 32225 (904) 221-2904
Yes! Please send information on pc-ditto.
Name_____
Address_____
City_____State____Zip_____